



## FACULTAD DE INFORMÁTICA

# TESINA DE LICENCIATURA

**TÍTULO:** Evaluación de la Calidad sobre una Aplicación Móvil

**AUTORES:** Ordóñez, Pablo Andrés

**DIRECTOR:** Mg. Esponda, Silvia

**CODIRECTOR:** -

**ASESOR PROFESIONAL:** -

**CARRERA:** Licenciatura en Sistemas

### Resumen

Esta tesina está clasificada como “*trabajo de investigación aplicada*” haciendo referencia a la *Evaluación de la Calidad sobre una Aplicación Móvil*. La calidad de software es una preocupación a la cual se dedican esfuerzos que valen la pena realizar. El objetivo primordial de toda organización es poseer productos de alta calidad para satisfacer las necesidades de sus clientes y, mejorar como organización frente a otras en el mercado competitivo.

El objetivo de la presente tesina es realizar un estudio evaluativo de la calidad de producto de software en una aplicación móvil concreta según la familia de normas ISO/IEC 25000 evaluando ciertas características fundamentales para productos desarrollados para dispositivos móviles, según métricas establecidas para poder realizar las mediciones y así detectar tanto sus ventajas como desventajas y poder alimentar un proceso de mejora continua.

### Palabras Clave

Aplicación Móvil,  
Evaluación, Métrica,  
Calidad, Producto,  
Software, ISO/IEC 25010,  
ISO/IEC 25023,  
ISO/IEC 25000, ISO/IEC  
25022, ISO/IEC 25040.

### Conclusiones

- Se aplicó la familia de las normas ISO/IEC 25000 para realizar una evaluación de calidad sobre una aplicación móvil real, obtener resultados, analizarlos y proponer acciones de mejora para el siguiente ciclo evaluativo, alimentando así, un proceso de mejora continua.
- Se adquirieron conocimientos acerca del proceso de evaluación de calidad de un producto de software, comprendiendo la importancia que se le da hoy en día a un producto de alta calidad, y las ventajas y beneficios que trae esto a las organizaciones desarrolladoras de estos productos.

### Trabajos Realizados

- Estudio sobre conceptos de calidad en el software y en aplicaciones móviles.
- Investigación sobre la familia de normas ISO/IEC 25000 (SQuaRE - System and Software Quality Requirements and Evaluation).
- Realización de una evaluación de calidad sobre una aplicación móvil real aplicando la familia de las normas ISO/IEC 25000.
- Realización de un reporte de la evaluación de calidad para descubrir debilidades y fortalezas del producto y así definir acciones de mejora para el mismo.

### Trabajos Futuros

- Estudio más detallado sobre el proceso de mejora continua y sus diferentes formas de integración con la norma de calidad estudiada (SQuaRE).
- Realización de un segundo ciclo de evaluación de calidad para la aplicación móvil seleccionada en el presente trabajo para continuar alimentando el proceso de mejora continua.
- Investigación sobre generación de métricas más específicas que las presentadas en la familia de normas ISO/IEC 25000 para las aplicaciones móviles.

# Evaluación de la Calidad sobre una Aplicación Móvil

**Pablo Andrés Ordóñez<sup>1</sup>**

Trabajo final para obtener el grado de  
Licenciado en Sistemas/Licenciatura en Sistemas

De la  
Facultad de Informática,  
Universidad Nacional de La Plata,  
La Plata, Buenos Aires, Argentina

Director: Mg. Esponda Silvia

La Plata, Agosto 2018

---

<sup>1</sup>Número de alumno: 11333/2 – Email: pablord89@hotmail.com

## Índice general

Índice de ilustraciones.....	6
Índice de tablas.....	7
Dedicatoria.....	8
Agradecimientos.....	9
CAPITULO 1.-INTRODUCCION.....	10
1.1.- Contexto.....	10
1.2.- Motivación.....	11
1.3.- Definición de objetivos.....	12
1.4.- Estructura del trabajo.....	12
CAPITULO 2.- CONCEPTOS DE CALIDAD EN EL SOFTWARE.....	14
2.1.- Definición de calidad.....	14
2.2.- Importancia de la calidad en el software.....	15
2.2.1.- Calidad del producto.....	16
2.2.2.- Dimensiones de la calidad del producto.....	16
2.2.3.- Medición de la calidad en el producto de software.....	20
2.2.4.- Calidad del proceso.....	21
2.3.- Familia de normas internacionales ISO/IEC 25000 (SQuaRE) .....	21
2.3.1.- Norma ISO/IEC 25000.....	21
2.3.2.- Objetivo.....	21
2.3.3.- Alcance.....	21
2.3.4.- Divisiones.....	22
2.3.4.1.- ISO/IEC 25010:2011, <i>Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models</i> .....	26
2.3.4.2.- ISO/IEC 25022:2014, <i>Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Measurement of quality in use</i> .....	29
2.3.4.3.- ISO/IEC 25023:2016, <i>Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Measurement of system and software product quality</i> .....	30
2.3.4.4.- ISO/IEC 25040:2011, <i>Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Evaluation process</i> .....	30
2.3.5.- Beneficios.....	34
2.3.6.- Vistas.....	34
2.3.7.- Relación con normas ISO/IEC 9126 e IRAM-ISO/IEC 14598.....	34
2.4.- Proceso de mejora continua.....	36
2.4.1.- Proceso de mejora continua aplicado a la familia de normas ISO/IEC 25000.....	37
2.5.- Conclusiones.....	38
CAPITULO 3.- CALIDAD EN APLICACIONES MOVILES.....	40
3.1.- Aplicación móvil.....	40
3.2.- Desarrollo de aplicaciones móviles.....	41
3.3.- Contexto móvil.....	41
3.4.- Calidad sobre aplicaciones móviles.....	42
3.5.- Pruebas de software sobre aplicaciones móviles.....	44
3.6.- Conclusiones.....	46
CAPITULO 4.- DESCRIPCION DEL PRODUCTO A EVALUAR.....	48
4.1.- Descripción general del producto de software.....	48
4.1.1.- Introducción.....	48
4.1.2.- Dominio y funcionamiento del producto.....	48
4.1.3.- Alcance del producto.....	48
4.1.4.- Objetivos.....	50
4.1.5.- Usuarios del producto.....	50
4.1.6.- Arquitectura.....	50
4.2.- Utilidad del producto en el mercado y para su evaluación de calidad.....	51
4.3.- Descripción detallada del producto de software.....	51
4.3.1.- Elementos propios del producto de software.....	51
4.3.2.- Pantallas del producto de software.....	51
4.3.3.- Interacción del producto con otros sistemas.....	58

4.3.4.- <i>Seguridad</i> del producto de software.....	58
4.4.- Organización desarrolladora del producto de software.....	59
4.4.1.- Descripción de la organización.....	59
4.4.2.- Infraestructura.....	59
4.5.- Conclusiones.....	59
CAPITULO 5.- REQUISITOS DE LA EVALUACION.....	60
5.1.- Propósito de la evaluación.....	60
5.2.- Motivación de la evaluación.....	60
5.3.- Tipo de producto de evaluar.....	60
5.4.- Modelo de calidad.....	60
5.4.1.- Selección de características a evaluar.....	61
5.4.1.1.- Importancia de la evaluación de la característica <i>usabilidad</i> para aplicaciones móviles.....	61
5.4.1.1.1.- Importancia de la evaluación de la subcaracterística inteligibilidad para aplicaciones móviles.....	61
5.4.1.1.2.- Importancia de la evaluación de la subcaracterística aprendizaje para aplicaciones móviles.....	61
5.4.1.1.3.- Importancia de la evaluación de la subcaracterística operabilidad para aplicaciones móviles.....	62
5.4.1.1.4.- Importancia de la evaluación de la subcaracterística protección frente a errores de usuario para aplicaciones móviles.....	62
5.4.1.1.5.- Importancia de la evaluación de la subcaracterística estética para aplicaciones móviles.....	62
5.4.1.1.6.- Importancia de la evaluación de la subcaracterística accesibilidad para aplicaciones móviles.....	62
5.4.1.2.- Importancia de la evaluación de la característica <i>seguridad</i> para aplicaciones móviles.....	62
5.4.1.2.1.- Importancia de la evaluación de la subcaracterística integridad para aplicaciones móviles.....	63
5.4.1.2.2.- Importancia de la evaluación de la subcaracterística autenticidad para aplicaciones móviles.....	63
5.5.- Requerimientos de calidad.....	63
5.6.- Grado de rigurosidad.....	65
5.7.- Grado de confidencialidad.....	65
5.8.- Conclusiones.....	66
CAPITULO 6.- ESPECIFICACION DE LA EVALUACION.....	67
6.1.- Alcance de la evaluación.....	67
6.2.- Selección de métricas.....	67
6.2.1.- Métricas seleccionadas para evaluar la <i>usabilidad</i> .....	67
6.2.2.- Métricas seleccionadas para evaluar la <i>seguridad</i> .....	69
6.2.3.- Descripción de las métricas seleccionadas.....	70
6.3.- Niveles de rating.....	73
6.3.1.- Niveles de rating para evaluar los requerimientos de calidad a través de las métricas.....	75
6.4.- Criterios de evaluación.....	78
6.4.1.- Criterios de evaluación de las subcaracterísticas de calidad.....	78
6.4.2.- Criterios de evaluación de las características de calidad.....	78
6.4.3.- Criterios de evaluación final de la calidad.....	78
6.5.- Conclusiones.....	79
CAPITULO 7.- PLAN DE EVALUACION.....	80
7.1.- Introducción.....	80
7.1.1.- Objetivo.....	80
7.1.2.- Descripción del producto a evaluar.....	80
7.1.3.- Requerimientos de la evaluación.....	80
7.1.4.- Alcance de la evaluación.....	80
7.2.- Características de calidad aplicables.....	81
7.3.- Lista de prioridades.....	81
7.4.- Objetivos de calidad.....	82
7.5.- Conclusiones.....	82
CAPITULO 8.- EJECUCION DE LA EVALUACION.....	83
8.1.- Ejecución de la evaluación de la característica <i>seguridad</i> .....	83

8.1.1.- Ejecución de la evaluación de la subcaracterística integridad.....	83
8.1.2.- Ejecución de la evaluación de la subcaracterística autenticidad.....	84
8.2.- Ejecución de la evaluación de la característica <i>usabilidad</i> .....	85
8.2.1.- Ejecución de la evaluación de la subcaracterística operabilidad.....	85
8.2.2.- Ejecución de la evaluación de la subcaracterística inteligibilidad.....	88
8.2.3.- Ejecución de la evaluación de la subcaracterística protección frente a errores de usuario.....	89
8.2.4.- Ejecución de la evaluación de la subcaracterística aprendizaje.....	91
8.2.5.- Ejecución de la evaluación de la subcaracterística accesibilidad.....	92
8.2.6.- Ejecución de la evaluación de la subcaracterística estética.....	93
8.3.- Conclusiones.....	94
CAPITULO 9.- REPORTE DE LA EVALUACION.....	95
9.1.- Identificaciones.....	95
9.1.1.- Reporte de evaluación.....	95
9.2.- Requerimientos de la evaluación.....	95
9.3.- Especificación de la evaluación.....	96
9.4.- Métodos de evaluación.....	96
9.5.- Resultados de la evaluación.....	96
9.5.1.- Resultados de la evaluación de las métricas de calidad.....	96
9.5.2.- Resultados de la evaluación de las subcaracterísticas de calidad.....	101
9.5.3.- Resultados de la evaluación de las características de calidad.....	104
9.5.4.- Conclusión final acerca de la calidad del producto de software.....	105
9.5.4.1.- Acciones de mejora propuestas.....	105
9.6.- Conclusiones.....	106
CAPITULO 10.- CONCLUSION.....	107
CAPITULO 11.- REFERENCIAS.....	109

## Índice de ilustraciones

Ilustración 1.....	17
Ilustración 2.....	18
Ilustración 3.....	18
Ilustración 4.....	19
Ilustración 5.....	20
Ilustración 6.....	23
Ilustración 7.....	26
Ilustración 8.....	31
Ilustración 9.....	35
Ilustración 10.....	36
Ilustración 11.....	37
Ilustración 12.....	40
Ilustración 13.....	43
Ilustración 14.....	44
Ilustración 15.....	50
Ilustración 16.....	52
Ilustración 17.....	52
Ilustración 18.....	53
Ilustración 19.....	54
Ilustración 20.....	54
Ilustración 21.....	55
Ilustración 22.....	56
Ilustración 23.....	56
Ilustración 24.....	57
Ilustración 25.....	57
Ilustración 26.....	58
Ilustración 27.....	58
Ilustración 28.....	74

## Índice de tablas

Tabla 1.....	22
Tabla 2.....	23
Tabla 3.....	24
Tabla 4.....	24
Tabla 5.....	25
Tabla 6.....	25
Tabla 7.....	38
Tabla 8.....	64
Tabla 9.....	65
Tabla 10.....	69
Tabla 11.....	70
Tabla 12.....	73
Tabla 13.....	77
Tabla 14.....	78
Tabla 15.....	81
Tabla 16.....	83
Tabla 17.....	84
Tabla 18.....	86
Tabla 19.....	86
Tabla 20.....	87
Tabla 21.....	88
Tabla 22.....	90
Tabla 23.....	90
Tabla 24.....	91
Tabla 25.....	92
Tabla 26.....	93
Tabla 27.....	95
Tabla 28.....	100
Tabla 29.....	101
Tabla 30.....	104
Tabla 31.....	104

# *Dedicatoria:*

*“A mi familia...”*



## Agradecimientos

Quiero agradecer, en principio, a toda mi familia que me apoyaron incondicionalmente durante toda la carrera: a mis padres que siempre me aconsejaron y guiaron, a mis hermanos que estuvieron ahí para ayudarme, a mis tíos y primos, siempre en las buenas y en las malas, y a mis abuelos, que hoy no están pero que siempre me dieron ánimos para concluir este gran logro en mi vida.

A mi novia, siguiéndome durante todo este tiempo, ayudándome en cada momento y en las decisiones difíciles a tomar, aportando ideas y estando para mí.

También agradecer a mis amigos que estuvieron siempre presentes haciendo que no baje los brazos y siga avocándome a esta hermosa carrera.

Agradecer también a mi directora, Silvia, que gracias a su capacidad profesional, su confianza, su amabilidad, su predisposición y su apoyo hicieron posible que yo pueda emprender este trabajo final y poder concluirlo, formándome como profesional en este largo camino que me espera.

---

## CAPITULO 1.- INTRODUCCION

---

### 1.1.- Contexto

En un mundo globalizado, en donde las organizaciones compiten constantemente, la calidad se convierte en un importante punto diferenciador, además de aumentar la satisfacción general del cliente, disminuir costos y optimizar los recursos. Cuando los productos de software son lanzados al mercado, se espera que tengan cierto grado de aceptación entre los usuarios, ese grado va a depender de las características particulares que cada usuario considere importantes. Desde el punto de vista de la Ingeniería de Software, una de las principales características, para ser exitoso entre los usuarios, es que sea de calidad. Resulta relevante para los desarrolladores de software poder medir esa calidad o realizar pruebas de calidad a las aplicaciones construidas, pero para poder medir se necesita saber qué es lo que hay que medir y cómo.

La obtención de un software con calidad implica la utilización de metodologías o procedimientos para el análisis, diseño, programación y pruebas del software que permitan unificar la filosofía de trabajo, con la finalidad de alcanzar una mayor confiabilidad, mantenibilidad y facilidad de prueba, a la vez que eleven la productividad, tanto para la labor de desarrollo como para el control de la calidad del software.

Según Mark C. Paulk: "La calidad de un producto es ampliamente gobernada por la calidad del proceso usado para construirlo" [B].

El software de alta calidad proporciona beneficios a la organización que lo produce y a la comunidad de usuarios finales. La organización que elabora el software obtiene valor agregado porque el software de alta calidad requiere un menor esfuerzo de mantenimiento, menos errores que corregir y poca asistencia al cliente. Esto permite que los ingenieros de software dediquen más tiempo a crear nuevas aplicaciones. El resultado final es:

- mayores utilidades por el producto de software,
- más rentabilidad cuando una aplicación apoya un proceso de negocios y
- mejor disponibilidad de información, que es crucial para el negocio.

Debido a que en los últimos años el uso de dispositivos móviles (teléfonos móviles, reproductores de audio portátil, asistentes personales digitales, navegadores gps, tablets, cámaras digitales, etc.) se ha incrementado de manera considerable, es importante disponer de metodologías y herramientas que permitan evaluar su calidad.

El uso de normas y estándares para lograr mayor competitividad, es cada vez más habitual en las organizaciones desarrolladoras de software, ya sea para la obtención de una certificación o simplemente lograr una mejora en el desarrollo del software a través del uso de buenas prácticas. La incorporación de las mismas en organizaciones desarrolladoras de software, tanto a nivel de realización de un producto, definición del proceso, o gestión, dejó de ser una aspiración para ser una necesidad de modo de aumentar la competitividad y lograr una mejora a través del uso de ellas.

Entre las normas asociadas a la calidad de producto de software se encuentra la familia de la norma ISO/IEC 25000, la cual se divide en varias normas, entre ellas la ISO/IEC 25010-System and software quality models, la ISO/IEC 25022-Measurement of quality in use, la ISO/IEC 25023-Measurement of system and software product

*quality* e ISO/IEC 25040-*Evaluation process*, las cuales se utilizarán durante todo este trabajo para poder realizar la evaluación de calidad del producto seleccionado. El objetivo general de la creación del estándar ISO/IEC 25000 SQuaRE (*System and Software Quality Requirements and Evaluation*) es organizar, enriquecer y unificar las series que cubren dos procesos principales: especificación de requisitos de calidad del software y evaluación de la calidad del software. Las características de calidad y sus mediciones asociadas pueden ser útiles no solamente para evaluar el producto software sino también para definir los requerimientos de calidad.

La norma ISO/IEC 25010-*System and software quality models* define las siguientes características y subcaracterísticas de calidad:

- Adecuación Funcional, subdividida en: Completitud, Corrección, Idoneidad.
- Eficiencia en el desempeño, subdividida en: Comportamiento temporal, Utilización de Recursos, Capacidad.
- *Usabilidad*, subdividida en: Inteligibilidad, Aprendizaje, Operabilidad, Protección a errores de usuario, Estética, Accesibilidad.
- Fiabilidad, subdividida en: Madurez, Disponibilidad, Tolerancia a Fallos, Capacidad de Recuperación.
- *Seguridad*, subdividida en: Confidencialidad, Integridad, No Repudio, Autenticidad, Responsabilidad.
- Mantenibilidad, subdividida en: Modularidad, Reusabilidad, Analizabilidad, Capacidad de ser modificado, Capacidad de ser probado.
- Portabilidad, subdividida en: Adaptabilidad, Facilidad de instalación, Intercambiabilidad.
- Compatibilidad, subdividida en: Coexistencia, Interoperabilidad. [G]

Al evaluar la calidad de un producto, se aplican métricas para medir sus características. La medición del software se ocupa de derivar un valor numérico o perfil para una característica de un componente, sistema o proceso de software. Al comparar dichos valores unos con otros, es posible extraer conclusiones sobre la calidad del software, o valorar la efectividad de los procesos, las herramientas y los métodos de software.

Este trabajo se enfoca en los distintos aspectos relacionados con la medición de características de calidad, teniendo en consideración aquellas más importantes para aplicaciones móviles.

Dichas aplicaciones son desarrolladas para ejecutarse en dispositivos móviles. El término móvil se refiere a poder acceder a los datos, las aplicaciones y los dispositivos desde cualquier lugar. Para desarrollar software de este tipo se debe tener en cuenta ciertas restricciones del hardware en estos dispositivos, como por ejemplo sus dimensiones, bajo poder de cómputo, escasa capacidad de almacenamiento, ancho de banda limitado, etc. Algunos ejemplos de aplicaciones móviles son: mapas y navegación, búsqueda, juegos, mensajería, aplicaciones empresariales.

### 1.2.- Motivación

Es muy importante para un producto de software mantener altos niveles de calidad ya que ayuda a mejorarlo en todos los aspectos y características, evitando riesgos que puedan llevar a que el producto decaiga en el mercado, y logrando aumentar la confianza y mejorar la visión que los usuarios tienen hacia el producto, haciendo que pueda competir con otros de manera continua y progresiva. De esta manera, se desea realizar la evaluación de calidad de una aplicación móvil en particular, evaluando ciertas características, con el objetivo de detectar falencias e iniciar un proceso de mejora continua de la calidad [A].

El término "app" (aplicación móvil) se ha puesto muy de moda en los últimos años, y más aún con la proliferación de los dispositivos móviles, pues, a medida que

transcurre el tiempo, son más las personas que las utilizan. Además, cada vez son más las PyMEs o empresas que desarrollan su aplicación móvil. El uso de este tipo de producto de software viene creciendo de manera exponencial, y, debido a esto, es de sumo interés en este trabajo realizar una evaluación de calidad a una app.

La ISO/IEC 25000 establece criterios para la especificación de requisitos de calidad de productos software, sus métricas y su evaluación, e incluye un modelo de calidad para organizar, enriquecer y unificar las definiciones de calidad de los clientes con los atributos en el proceso de desarrollo. Uno de los principales beneficios que ofrece la norma es que alinea los objetivos del software con las necesidades reales, evitando ineficiencias y maximizando la rentabilidad y calidad del producto de software [E] .

### 1.3.- Definición de los objetivos

El objetivo general de este trabajo es realizar un estudio evaluativo de la calidad de producto de software en una aplicación móvil concreta según la familia ISO/IEC 25000 (SQuaRE: System and Software Quality Requirements and Evaluation), tomando, específicamente, ISO/IEC 25010-*System and software quality models*, ISO/IEC 25022-*Measurement of quality in use*, ISO/IEC 25023-*Measurement of system and software product quality* e ISO/IEC 25040-*Evaluation process* evaluando ciertas características fundamentales para productos desarrollados para dispositivos móviles, según métricas establecidas por estos estándares y así detectar tanto sus ventajas como desventajas y poder promover un proceso de mejora continua. Específicamente:

1. Estudiar dentro de la familia de las normas ISO/IEC 25000, las normas ISO/IEC 25010, ISO/IEC 25022, ISO/IEC 25023 e ISO/IEC 25040.
2. Estudiar conceptos de calidad del software en general, y, en particular, su utilización en aplicaciones móviles (app).
3. Definir características y métricas aplicables a app's en general.
4. Realizar una evaluación de calidad completa y detallada de la aplicación móvil CargoX desarrollada por la empresa Sontra.

### 1.4.- Estructura del trabajo

En el **CAPITULO 2: CALIDAD EN EL SOFTWARE**, se presentan definiciones de calidad de software y su importancia, así como también se describe la calidad del producto de software orientado a aplicaciones móviles. Se detalla gran parte de la familia de la norma ISO/IEC 25000: ISO/IEC 25010-*System and software quality models*, ISO/IEC 25022-*Measurement of quality in use*, ISO/IEC 25023-*Measurement of system and software product quality* e ISO/IEC 25040-*Evaluation process*.

En el **CAPITULO 3.- CALIDAD EN APLICACIONES MOVILES** se presentan conceptos como el de aplicación móvil, realizando un análisis sobre las distintas formas de asegurar y controlar la calidad en aplicaciones móviles, mencionando las pruebas de software más importantes que se realizan hoy en día sobre app's para mejorar la calidad y explicando las características más importantes a tener en cuenta para este tipo de aplicaciones.

En el **CAPITULO 4.- DESCRIPCION DEL PRODUCTO A EVALUAR**, se describe la aplicación móvil (producto de software) elegida para el presente trabajo, así como también se describe brevemente la empresa de software encargada de desarrollar y mantener esta aplicación móvil.

En el **CAPITULO 5.- REQUERIMIENTOS DE LA EVALUACION**, se especifican los requerimientos de la evaluación de calidad sobre el producto de software elegido.

Aquí se definen los objetivos de la misma, el comportamiento del producto y las características que se someten a evaluación, en este caso la *usabilidad* y la *seguridad*.

En el *CAPITULO 6.- ESPECIFICACION DE LA EVALUACION* se define el alcance de la evaluación, y se selecciona las métricas y criterios de evaluación de las mismas, con el fin de satisfacer los objetivos de la evaluación.

Una vez especificada la evaluación, comienza el diseño de la misma.

El *CAPITULO 7.- PLAN DE EVALUACION* se describe los métodos, herramientas y la organización de las acciones del evaluador.

En el *CAPITULO 8.- EJECUCION DE LA EVALUACION* se presentan los cálculos de las métricas seleccionadas para cada subcaracterística, y luego para cada característica de calidad, y llegar a una conclusión final y, proponer acciones de mejora en el *CAPITULO 9.- REPORTE DE LA EVALUACION*, en el cual también se muestra el reporte completo y detallado de los resultados obtenidos con comentarios y observaciones.

Finalmente, en el *CAPITULO 10.- CONCLUSIÓN* se presentan la conclusiones finales del trabajo, y en el *CAPITULO 11.- REFERENCIAS* utilizadas en la presente tesina.

---

## CAPITULO 2.- CALIDAD EN EL SOFTWARE

---

### 2.1.- Definición de calidad

La calidad del software es un concepto complejo que no es directamente comparable con la calidad de un producto en general. El software se ha convertido hoy en día en uno de los principales objetivos estratégicos de las organizaciones debido a que, cada vez mas, los procesos de las organizaciones dependen del buen funcionamiento de los sistemas de software.

El significado de la palabra calidad genera mucha confusión, quizás por la multidimensionalidad del propio concepto.

Por este motivo, pueden darse varias definiciones que son muy similares entre ellas, para el concepto de calidad:

- el glosario de estándares de computación IEEE Std. 610 – 1991, define la calidad del software como:

*“el grado con el que un sistema, componente o proceso cumple los requerimientos especificados y las necesidades o expectativas del cliente o usuario.”*

- las principales normas internacionales definen la calidad como:

*“el grado en el que un conjunto de características inherentes cumple con los requisitos.” (ISO 9000).*

*“la totalidad de características de una entidad que influyen en la capacidad para satisfacer necesidades explícitas o implícitas”. (ISO 8402).*

- el diccionario de la Real Academia Española lo define como:

*“propiedad o conjunto de propiedades inherentes a algo, que permiten juzgar su valor.”*

Algunas consideraciones o características a tener en cuenta sobre la calidad de software:

- Es un concepto relativo y comparativo
- Es un concepto multidimensional:
  - referida a diversas cualidades (como funcionalidad, costo, mantenibilidad, etc).
  - sujeta a restricciones (como el presupuesto disponible para invertir en obtener o lograr determinado nivel de calidad, o al plazo de fabricación en cuanto al tiempo que puede insumir fabricar algo con determinado nivel de calidad).
  - ligada a ciertos compromisos.
- Posee algunos aspectos subjetivos (existen cualidades cuya evaluación solo puede ser subjetiva) y otros objetivos (pueden medirse aspectos que influyen en la calidad).

A su vez, la calidad posee diferentes visiones:

- Visión trascendental: es aquella visión que puede ser reconocida pero no definida.

- Visión del usuario: es el grado de adecuación al propósito. Es aquel software que satisface las necesidades de los usuarios, que se basa en tareas del usuario a veces muy dependiente del contexto, apoyado por modelos de confiabilidad, modelos de desempeño y laboratorios de usabilidad.
- Visión del productor: Es aquella conformidad que se obtiene con la especificación de dicho producto.
- Visión del producto: está ligada a características inherentes del mismo. Por lo tanto las características internas determinan la calidad del mismo.
- Visión basada en valor: es aquella que está basada en; ¿Cuánto el cliente está dispuesto a pagar?.

Cuando no hay calidad en el software pueden ocurrir situaciones no deseadas como:

- Programas que no hacen exactamente lo que se espera.
- Proyectos que no terminan nunca.
- Sistemas informáticos que no se utilizan por la dificultad de su manejo.
- Productos de software que son imposibles de mantener cuando desaparece la persona o personas que lo desarrollaron.
- Software poco seguro.

## 2.2.- Importancia de la calidad en el software

La calidad del software puede parecer un concepto alejado de la vida diaria de la mayoría de las personas, pero nada más lejos de la realidad. Cuando en nuestra PC aparece un mensaje de error, estamos ante un problema de calidad del software; cuando un fallo en el sistema de gestión aeroportuaria provoca retrasos, pérdidas de maletas o inutiliza pantallas de información, estamos ante un problema de calidad del software; cuando en un restaurante se bloquean los terminales de cobro y pedidos, estamos ante un problema de calidad del software; cuando nuestro dispositivo móvil no inicia una aplicación o ésta se cierra inesperadamente, estamos también en presencia de un problema de calidad del software.

Es muy probable que hayamos sufrido los efectos de estos problemas de calidad en forma de retrasos, pérdidas de tiempo o dinero, etc. Lamentablemente, estos problemas pueden ser mucho más graves si afectan a sistemas críticos; es decir, aquellos cuya falla puede provocar graves pérdidas económicas o problemas ambientales o sociales e, incluso, la pérdida de vidas humanas. Un ejemplo bien documentado se produjo con el lanzamiento en 1996 del primer cohete Ariane 5 de la Agencia Espacial Europea. Este ingenio que costó 10 años y 7.000 millones de euros desarrollar, explotó por un defecto en el software de control interno antes de que pasara un minuto de vuelo [Q].

El correcto funcionamiento del software es indispensable para el éxito de los negocios e incluso para la seguridad humana, por tanto se vuelve importante la realización de una evaluación de calidad sobre el producto de software. Esta evaluación se lleva a cabo definiendo características de calidad del producto de acuerdo al propósito de uso del mismo.

El software de alta calidad proporciona beneficios a la organización que lo produce y a los usuarios finales. La organización que elabora el software obtiene valor agregado porque el software de alta calidad requiere un menor esfuerzo de mantenimiento, menos errores que corregir y poca asistencia al cliente. Esto permite que los ingenieros de software dediquen más tiempo a crear nuevas aplicaciones. Los usuarios obtienen valor agregado porque la aplicación provee una capacidad útil en forma tal que agiliza algún proceso de negocios. El resultado final es:

- 1) el producto de software puede adquirir mayores utilidades por parte de los usuarios,
- 2) más rentabilidad cuando una aplicación apoya un proceso de negocios y



3) mejor disponibilidad de información, que es crucial para el negocio [C].

Un producto de software (como por ejemplo una aplicación móvil como Skype o Whatsapp), con deficiente calidad provocará principalmente desconfianza y malestar en el cliente y/o usuario, traducándose esto en costos para la empresa que desarrolló el producto, dado que tendrá que invertir tiempo en re-trabajo para solucionar los inconvenientes del software y puede hasta llegar a perder clientes. Además, según el contexto de uso del producto, los riesgos de inconvenientes en el mismo pueden llegar a ser graves e incluso catastróficos. Por otro lado producir software de calidad conlleva a beneficios tanto para el usuario como para el proveedor, dado que el usuario se verá conforme y confiado en el producto, ya que a mayor nivel de calidad será mayor el nivel de confianza del usuario en éste, y el proveedor irá afianzando su experiencia en el desarrollo, hará crecer el producto mejorándolo continuamente y se verá beneficiado económicamente porque la inversión en su producto será no tanto en re-trabajo para solucionar inconvenientes (los cuales siempre existirán) sino para expandir el producto y su mercado.

La calidad del Software se divide en la calidad del producto obtenido y la calidad del proceso de desarrollo. Las mismas son dependientes ya que para lograr calidad en un producto debe haber calidad en su proceso de desarrollo. Las metas que se establezcan para la calidad del producto van a determinar los objetivos del proceso de desarrollo, ya que la calidad del primero va a depender, entre otros aspectos, de la calidad del mencionado proceso. Los requerimientos de calidad más significativos del proceso de software son:

- Que produzca los resultados esperados
- Que los resultados estén basados en una correcta definición.
- Que los resultados sean mejorados en función de los objetivos de negocio.

### 2.2.1.- Calidad del producto

La calidad del producto, junto con la calidad del proceso, es uno de los aspectos más importantes en el desarrollo de Software. Relacionada con la calidad del producto, recientemente ha aparecido la familia de normas ISO/IEC 25000, que proporciona una guía para el uso de la nueva serie de estándares internacionales llamada Requisitos y Evaluación de Calidad de Productos de Software (*SQuaRE - System and Software Quality Requirements and Evaluation*), tomando como producto de software para este trabajo, una aplicación móvil.

El producto es un bien tangible que es el resultado de un proceso. Aunque el software tiene aspectos intangibles, un producto software es sin embargo un bien en sí mismo e incluye sus documentos asociados.

Existen dos enfoques posibles que hacen referencia a la calidad del producto de software:

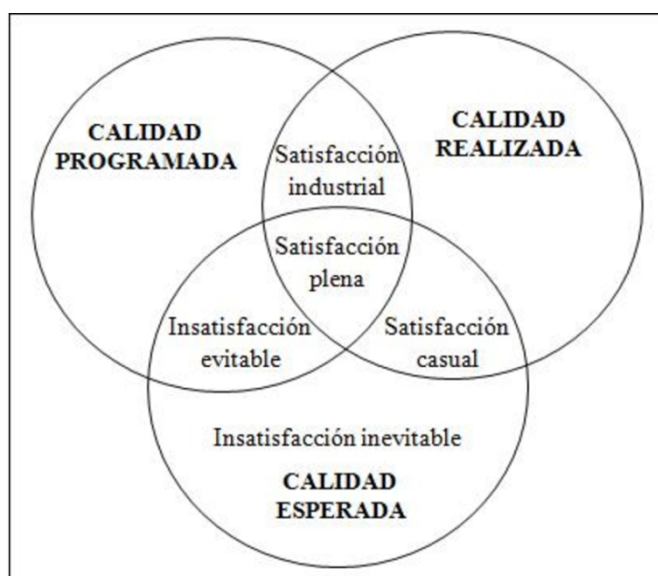
- Calidad funcional: refleja en qué medida el software cumple con o se ajusta a un determinado diseño, basado en requerimientos funcionales. Éstos abarcan las actividades del software que involucran procesamiento de datos de entrada.
- Calidad estructural: refleja en qué medida el software cumple con los requerimientos no funcionales, como rendimiento, capacidad de mantenimiento o escalabilidad.

### 2.2.2.- Dimensiones de la calidad del producto

Para alcanzar la satisfacción plena del cliente deben coincidir la calidad programada o diseñada, la calidad realizada y la calidad esperada:



- 1.La calidad programada o diseñada es la que la empresa pretende obtener (calidad prevista), y que se plasma en las especificaciones de diseño del producto, con el fin de responder a las necesidades del cliente.
- 2.La calidad realizada es la obtenida tras la producción, y tiene que ver con el grado de cumplimiento de las características de calidad del producto tal como se plasmaron en las especificaciones de diseño.
- 3.La calidad esperada, necesaria o concertada es la necesitada por el cliente según se manifiesta en sus necesidades y expectativas.



**Ilustración 1 – Calidad ideal**

El concepto de calidad ideal se basa en conseguir que los tres círculos sean concéntricos, cuando la satisfacción del cliente es plena (véase ilustración 1).

La concepción global de calidad puede expandirse más aún si distinguimos entre la calidad esperada y la calidad latente o sorpresiva. La calidad latente es aquella que el cliente no esperaba encontrar en el producto porque no figuraba en la compra convenida, pero que, una vez recibida, le satisface porque se anticipa a una necesidad subyacente. Aquí podemos distinguir tres situaciones (véase ilustración 2):

1. Insatisfacción del cliente. La calidad realizada no alcanza a cubrir las necesidades del cliente, y lógicamente menos aún colma sus expectativas. Son las circunstancias que propician la pérdida de mercado y la pérdida de reputación.

2. Satisfacción del cliente. La calidad realizada está a la altura de la calidad esperada, respondiendo el producto a las necesidades del cliente. En la medida que el comprador no recibe más de lo que espera, su fidelización estará condicionada por la oferta de la competencia. La reputación de la empresa no se ve afectada por contemplarla el cliente como un suministrador normal. El “cero defectos” es lo menos que se puede pedir.

3. Entusiasmo del cliente. La calidad realizada cubre no sólo las necesidades sino también las expectativas del cliente. La empresa ofrece una calidad superior a la calidad latente que existe en la mente del consumidor, que la visualiza como un ofertante extraordinario. Es el mejor camino para lograr la lealtad del cliente.



Ilustración 2 – Calidad en el cliente

Se puede entonces contraponer tres modelos de empresa, según el concepto de calidad que en ella domine (véase ilustración 3):

1. Empresa acomodaticia. Su pensamiento en calidad está dominado por el nivel genérico de las especificaciones de referencia en la industria.
2. Empresa cumplidora. Su pensamiento en calidad está orientado a la satisfacción de las necesidades del cliente, dominando pues el logro de la calidad esperada.
3. Empresa enriquecida. Su pensamiento en calidad está orientado a superar las necesidades de los clientes, proporcionando un nivel de calidad potencial adicional que satisfaga o rebase sus expectativas. Está pues dominado por el logro de un nivel enriquecido.

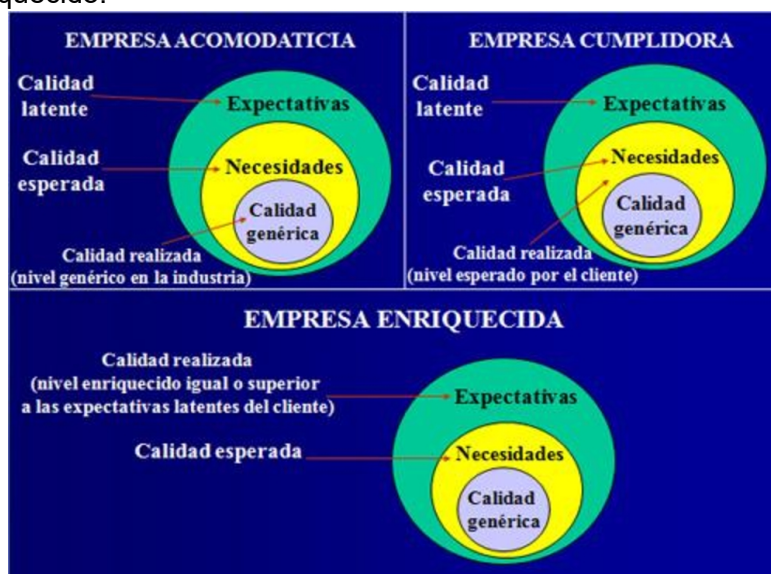


Ilustración 3 – Calidad en empresas

El concepto de calidad total del producto sugiere que la calidad ha de estar presente en todas las fases de su ciclo de vida, desde el diseño hasta el servicio postventa. Para optimizar la creación de valor para el cliente, la empresa debe decidir anticipadamente qué calidad del producto planificar, lograr y transmitir al cliente. Por consiguiente, debe identificar a través de la investigación del mercado las características que el producto debe reunir para satisfacer los requisitos de los clientes (calidad como aptitud para el uso). Dichas características se deben trasladar a especificaciones del producto, siendo su fabricación responsable de que el producto elaborado cumpla los requisitos de diseño (calidad como conformidad con especificaciones). El producto, tras su comercialización y venta a un precio que refleje el valor que tiene para el cliente (calidad como valor, en la acepción primigenia), satisfará al comprador si está a la altura de sus expectativas (calidad como satisfacción de expectativas). Pero, tanto en el diseño como en la comercialización, la empresa deberá tener en cuenta que la calidad, es algo que se percibe subjetivamente (calidad como excelencia).

Un modelo más operativo de lo que significa la calidad total del producto cuando se combinan las distintas perspectivas puede alcanzarse distinguiendo sus dimensiones. Los factores que determinan la percepción de calidad por el cliente pueden clasificarse en 6 dimensiones (véase ilustración 4).

- Calidad de concepción.
- Calidad de diseño.
- Calidad de conformidad.
- Calidad de entrega.
- Calidad percibida
- Calidad de servicio



Ilustración 4 – Dimensiones de la calidad

Estas dimensiones de la calidad son en unos casos independientes, mientras que en otros casos están fuertemente relacionadas. En su conjunto, todas ellas forman

el soporte para el desarrollo de la Gestión de la Calidad, siendo conceptos referidos a fases de actividades interdependientes. [K]

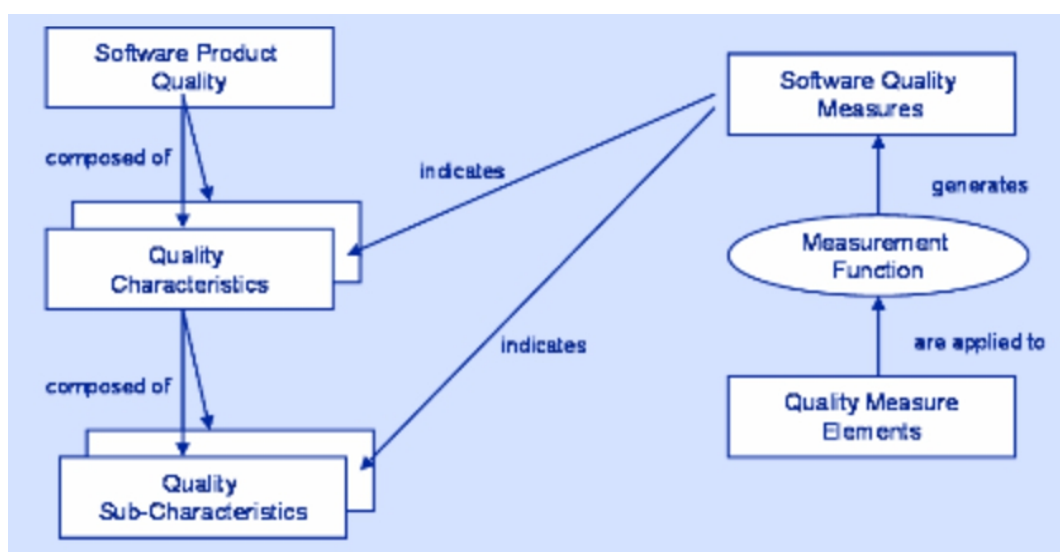
### 2.2.3.- Medición de la calidad en el producto de software

Una característica (denominado así por la norma ISO/IEC 25000) se define como una propiedad física o abstracta medible perteneciente a una entidad (producto de software, o alguno de sus componentes).

Las características se miden por medio de métricas, una métrica es un método de medición y su escala. La medición se define como el uso de un procedimiento para asignar un valor (puede ser un número o una categoría) de una escala a una característica de una entidad, y la escala es un conjunto de valores con propiedades definidas, es el universo donde se encuentra el conjunto de posibles resultados de la métrica. La medida se define como un número o categoría asignado a una característica de una entidad por medio de una medición.

La medición del software se ocupa de derivar un valor numérico o perfil para una característica de un componente, sistema o proceso de software. Al comparar dichos valores unos con otros, y con los estándares que se aplican a través de una organización, es posible extraer conclusiones sobre la calidad del software, o valorar la efectividad de los procesos, las herramientas y los métodos de software [D]. Las características del producto se agrupan en subcaracterísticas de calidad, y a su vez, éstas en características de calidad. Estas agrupaciones de las características se definen de acuerdo al factor común en cuanto al aspecto de las propiedades del producto que se pueden dimensionar a través de ellos (funcionalidad, confiabilidad, seguridad, etc.).

El grado o nivel de calidad que posee un producto es el resultado de aplicarle diferentes métricas para medir ciertos aspectos o características de calidad y puntuar los valores medidos. Estas puntuaciones se combinan para determinar una puntuación final que indica el nivel de calidad del producto, y tanto la puntuación final como las intermedias se someten a comparaciones ya sea contra criterios de evaluación preestablecidos, los cuales establecen rangos de valores inaceptables, mínimamente aceptables y planeados, como contra la puntuación obtenida por otros productos para los cuales se haya definido un esquema de evaluación similar.



**Ilustración 5: Modelo de Referencia de Medición de la Calidad del Producto Software, según la ISO/IEC 25000.**



### 2.2.4.- Calidad del proceso

El proceso se define como el conjunto de actividades, métodos, prácticas y transformaciones usadas para desarrollar y mantener los productos de software y sus productos de trabajo asociados.

La calidad de los procesos de desarrollo de software de una organización es uno de sus pilares más importantes, tanto por ser un elemento diferenciador para sus clientes, como por la importancia interna. Para que una organización pueda asegurar la calidad de sus procesos es necesario llevar a cabo un proceso de evaluación y mejora de los mismos, que permita alinear el desarrollo de la empresa a las mejores prácticas de la Ingeniería del Software.

La evaluación y mejora de la calidad de los procesos de desarrollo software es una práctica ampliamente extendida, cuyo objetivo es controlar la calidad del software a través de la institucionalización y mejora continua de los procesos que se utilizan para su desarrollo. Para ello, a lo largo de los últimos años han ido surgiendo diferentes modelos para evaluar y mejorar la capacidad de los procesos de desarrollo software y la madurez de las fábricas de software, entre los que se puede destacar CMM, CMMI, ISO/IEC 15504, ISO/IEC 9001 (bajo la guía 90003), entre otros.

### 2.3.- Familia de normas internacionales ISO/IEC 25000 (SQuaRE)

#### 2.3.1.- Norma ISO/IEC 25000

La familia ISO/IEC 25000 es el resultado de la evolución de otras normas anteriores, especialmente de las normas ISO/IEC 9126, que describe las particularidades de un modelo de calidad del producto software, e ISO/IEC 14598, que aborda el proceso de evaluación de productos software (ver relación entre estas normas más adelante). Esta familia de normas ISO/IEC 25000 se encuentra compuesta por cinco divisiones.

La norma ISO/IEC 25000 establece criterios para la especificación de requisitos de calidad de productos software, sus métricas y su evaluación, e incluye un modelo de calidad para unificar las definiciones de calidad de los clientes con las características en el proceso de desarrollo, es decir, es una familia de normas que tiene como objetivo la creación de un marco de trabajo común para evaluar la calidad del producto de software [F].

#### 2.3.2.- Objetivo

“La calidad del producto, junto con la calidad del proceso, es uno de los aspectos más importantes actualmente en el desarrollo de Software. La familia de normas ISO/IEC 25000, que proporciona una guía para el uso de la nueva serie de estándares internacionales llamada Requisitos y Evaluación de Calidad de Productos de Software (*SQuaRE - System and Software Quality Requirements and Evaluation*).

ISO/IEC 25000 constituye una serie de normas basadas en ISO/IEC 9126 y en ISO/IEC 14598 cuyo objetivo principal es guiar el desarrollo de los productos de software mediante la especificación de requisitos y evaluación de características de calidad.” [F]

#### 2.3.3.- Alcance

ISO/IEC 25000 va dirigido a las empresas de software, independiente de su tamaño o volumen.

## 2.3.4.- Divisiones

SQuaRE está formada por las siguientes divisiones:

Divisiones	Descripción
ISO/IEC 2500n - <i>Quality Management Division</i>	División de gestión de calidad. Los estándares que forman esta división define todos los modelos comunes, términos y referencias a los que se alude en las demás divisiones de SQuaRE.
ISO/IEC 2501n - <i>Software Quality Model</i>	División del modelo de calidad. El estándar que conforma esta división presenta un modelo de calidad detallado, incluyendo características para la calidad interna, externa y en uso.
ISO/IEC 2502n - <i>Measurement of Software Quality</i>	División de mediciones de calidad. Los estándares pertenecientes a esta división incluyen un modelo de referencia de calidad del producto software, definiciones matemáticas de las métricas de calidad y una guía práctica para su aplicación. Presenta aplicaciones de métricas para la calidad de software interna, externa y en uso.
ISO/IEC 2503n - <i>Software Quality Requirements</i>	División de requisitos de calidad. Los estándares que forman parte de esta división ayudan a especificar los requisitos de calidad. Estos requisitos pueden ser usados en el proceso de especificación de requisitos de calidad para un producto software que va a ser desarrollado ó como entrada para un proceso de evaluación. El proceso de definición de requisitos se guía por el establecido en la norma ISO/IEC 15288 (ISO, 2003).
ISO/IEC 2504n - <i>Software Quality Evaluation</i>	División de evaluación de la calidad. Estos estándares proporcionan requisitos, recomendaciones y guías para la evaluación de un producto software, tanto si la llevan a cabo evaluadores, como clientes o desarrolladores.
ISO/IEC 25050-25099	Estándares de extensión SQuaRE. Incluyen requisitos para la calidad de productos de software "Off-The-Self" y para el formato común de la industria (CIF) para informes de usabilidad.

**Tabla 1 – Divisiones de SQuaRE**

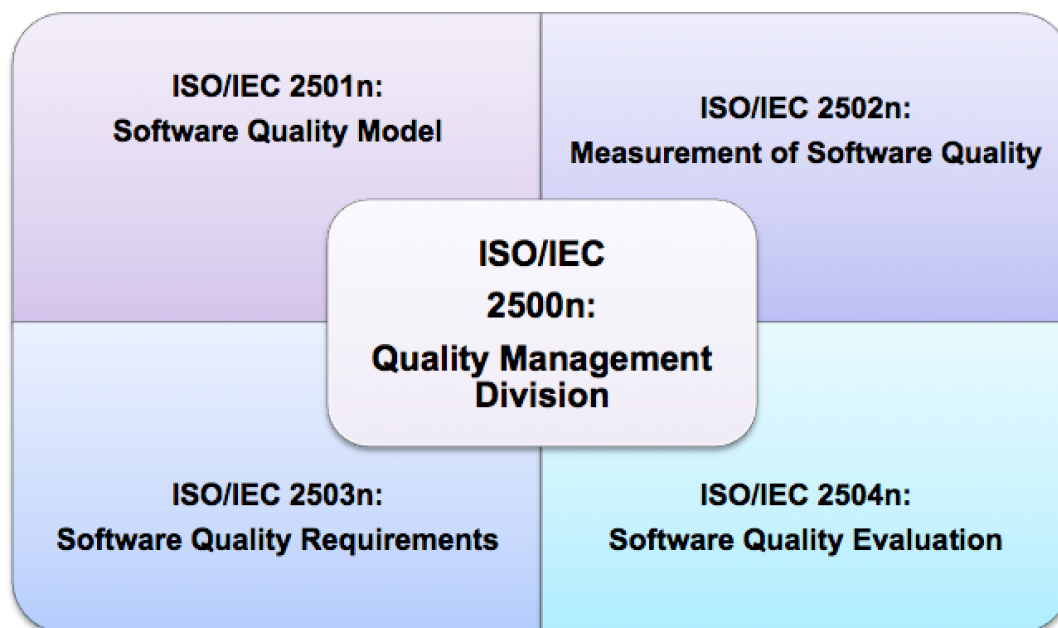


Ilustración 6 – Divisiones de calidad según SQuaRE

A continuación se mencionan las divisiones de la norma ISO/IEC mencionadas anteriormente en la tabla 1:

La ISO/IEC 2500n está formada por:

Divisiones	Descripción
ISO/IEC 25000 - <i>Guide to SQuaRE</i>	Contiene el modelo de la arquitectura de SQuaRE, la terminología de la familia, un resumen de las partes, los usuarios previstos y las partes asociadas, así como los modelos de referencia.
ISO/IEC 25001 - <i>Planning and Management</i>	Describe el soporte necesario para la gestión en la evaluación y especificación de requisitos de calidad.

Tabla 2- Divisiones de la ISO/IEC 2500n

La ISO/IEC 2501n se divide en:

Divisiones	Descripción
ISO/IEC 25010 - <i>System and software quality models</i>	Describe el modelo de calidad para el producto software y para la calidad interna, externa y en uso. Esta descripción se hace descomponiendo cada tipo de calidad en características, aunque el nivel de descomposición es mayor para la calidad en uso que para la interna o externa. En esta norma se presentan las características y subcaracterísticas de calidad frente a las cuales evaluar el producto software.

ISO/IEC 25012 - <i>Data Quality model</i>	Define un modelo general para la calidad de los datos, aplicable a aquellos datos que se encuentran almacenados de manera estructurada y forman parte de un Sistema de Información.
---	---

**Tabla 3- Divisiones de la ISO/IEC 2501n**

La ISO/IEC 2502n se divide en:

Divisiones	Descripción
ISO/IEC 25020 - <i>Measurement reference model and guide</i>	Presenta una explicación introductoria y un modelo de referencia común para las primitivas, medidas de calidad interna, externas y en uso. También contiene anexos informativos que abordan los siguientes temas: <ul style="list-style-type: none"> <li>● criterios para seleccionar medidas de calidad de software y elementos de medida de calidad,</li> <li>● demostrar la validez predictiva y evaluar la confiabilidad de la medición,</li> <li>● y un formato de ejemplo para documentar las medidas de calidad del software.</li> </ul>
ISO/IEC 25021 - <i>Quality measure elements</i>	Define y especifica un conjunto recomendado de métricas base y derivadas que puedan ser usadas a lo largo de todo el ciclo de vida del desarrollo software como entrada en el proceso de medida de la calidad interna, externa y en uso.
ISO/IEC 25022 - <i>Measurement of quality in use</i>	Define específicamente las métricas para realizar la medición de la calidad en uso del producto.
ISO/IEC 25023 - <i>Measurement of system and software product quality</i>	Define específicamente las métricas para realizar la medición de la calidad de productos y sistemas software.
ISO/IEC 25024 - <i>Measurement of data quality</i>	Define específicamente las métricas para realizar la medición de la calidad de datos.

**Tabla 4- Divisiones de la ISO/IEC 2502n**



La ISO/IEC 2503n está constituida por:

Divisiones	Descripción
ISO/IEC 25030 - <i>Quality requirements</i>	Provee de un conjunto de recomendaciones para realizar la especificación de los requisitos de calidad del producto software. Estos requisitos pueden utilizarse en la licitación de contratos de desarrollo pero también como entradas en el proceso de evaluación. Cabe destacar, que según el propio estándar, el proceso de definición de requisitos se corresponden con procesos técnicos definidos en el estándar ISO/IEC 15288 - <i>Procesos del ciclo de vida</i> .

**Tabla 5- Divisiones de la ISO/IEC 2503n**

La ISO/IEC 2504n se divide en:

Divisiones	Descripción
ISO/IEC 25040 - <i>Quality evaluation reference model and guide</i>	Describe los requisitos generales a cumplir en la especificación y evaluación de la calidad de un software. Proporciona una base para la evaluación de la calidad de un software. Especifica los requisitos que deben cumplir los métodos de evaluación y medida de un producto de software.
ISO/IEC 25041 - <i>Evaluation modules</i>	Describe la estructura y contenido de la documentación que debe describir los módulos de evaluación.
ISO/IEC 25042 - <i>Process for developers</i>	Proporciona requisitos y recomendaciones de carácter práctico para la implementación de la evaluación cuando esta se da en paralelo con el desarrollo.
ISO/IEC 25043 - <i>Process for acquirers</i>	Describe requisitos y recomendaciones para la medida y evaluación sistemática de productos de software comerciales, productos desarrollados a medida, o productos a modificar bajo contrato.
ISO/IEC 25044 - <i>Process for evaluators</i>	Detalla requisitos y recomendaciones para la evaluación de software de forma que dicha evaluación sea fiable y comprensible.
ISO/IEC 25045 - <i>Evaluation module for recoverability</i>	Define un módulo para la evaluación de la subcaracterística recuperabilidad (recoverability).

**Tabla 6 -Divisiones de la ISO/IEC 2504n**

La división de extensión de SQuaRE (ISO/IEC 25050 a ISO/IEC 25099) se reserva para la norma o informes técnicos que aborden dominios de aplicación específicos o que puedan ser utilizados para complementar otras normas de la familia SQuaRE.

En el presente trabajo se abordarán principalmente las normas ISO/IEC 25010-*System and software quality models*, ISO/IEC 25022-*Measurement of quality in use*,

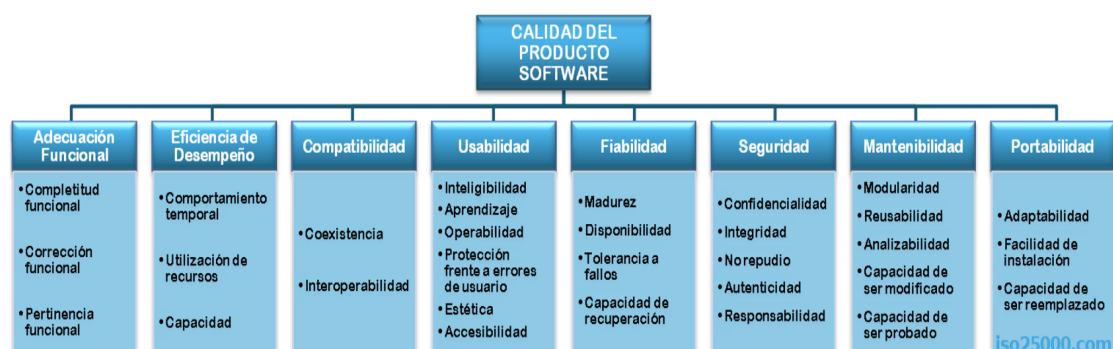
ISO/IEC 25023-*Measurement of system and software product quality* e ISO/IEC 25040 para realizar la evaluación de la calidad sobre una aplicación móvil a partir de métricas que aportan estas normas, realizando las mediciones correspondientes para obtener un producto de mayor calidad. Por este motivo se definen en detalle a continuación solo estas normas, y no todas las de la familia de la norma ISO/IEC 25000.

### 2.3.4.1.- ISO/IEC 25010:2011, *Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models*

El modelo de calidad representa la piedra angular en torno a la cual se establece el sistema para la evaluación de la calidad del producto de software. En este modelo se determinan las características de calidad que se van a tener en cuenta a la hora de evaluar las propiedades de un producto software<sup>2</sup> determinado.

La calidad del producto software se puede interpretar como el grado en que dicho producto satisface los requisitos de sus usuarios aportando de esta manera un valor. Son precisamente estos requisitos (funcionalidad, rendimiento, *seguridad*, mantenibilidad, etc.) los que se encuentran representados en el modelo de calidad, el cual categoriza la calidad del producto en características y subcaracterísticas.

El modelo de calidad del producto definido por la ISO/IEC 25010 (abordado con más detalle en la sección 5.4.- *Modelo de calidad*) se encuentra compuesto por las ocho características de calidad que se muestran en la ilustración 7.



**Ilustración 7: Características y subcaracterísticas de calidad según SQuaRE. [F]**

#### 1. Adecuación Funcional

Representa la capacidad del producto software para proporcionar funciones que satisfacen las necesidades declaradas e implícitas, cuando el producto se usa en las condiciones especificadas. Esta característica se subdivide a su vez en las siguientes subcaracterísticas:

- **Completitud funcional:** grado en el cual el conjunto de funcionalidades cubre todas las tareas y los objetivos del usuario especificados.
- **Corrección funcional:** capacidad del producto o sistema para proveer resultados correctos con el nivel de precisión requerido.
- **Pertinencia funcional:** capacidad del producto software para proporcionar un conjunto apropiado de funciones para tareas y objetivos de usuario especificados.

2. Nota al lector: incluya dentro de producto de software a aplicaciones móviles.

## 2. Eficiencia de desempeño o Rendimiento

Esta característica representa el desempeño relativo a la cantidad de recursos utilizados bajo determinadas condiciones. Esta característica se subdivide a su vez en las siguientes subcaracterísticas:

- Comportamiento temporal: los tiempos de respuesta y procesamiento y los ratios de *throughput* de un sistema cuando lleva a cabo sus funciones bajo condiciones determinadas en relación con un banco de pruebas (*benchmark*) establecido.
- Utilización de recursos: las cantidades y tipos de recursos utilizados cuando el software lleva a cabo su función bajo condiciones determinadas.
- Capacidad: grado en que los límites máximos de un parámetro de un producto o sistema software cumplen con los requisitos.

## 3. Compatibilidad

Capacidad de dos o más sistemas o componentes para intercambiar información y/o llevar a cabo sus funciones requeridas cuando comparten el mismo entorno hardware o software. Esta característica se subdivide a su vez en las siguientes subcaracterísticas:

- Coexistencia: capacidad del producto para coexistir con otro software independiente, en un entorno común, compartiendo recursos comunes sin detrimento.
- Interoperabilidad: capacidad de dos o más sistemas o componentes para intercambiar información y utilizar la información intercambiada.

## 4. Usabilidad

Capacidad del producto software para ser entendido, aprendido, usado y resultar atractivo para el usuario, cuando se usa bajo determinadas condiciones. Esta característica se subdivide a su vez en las siguientes subcaracterísticas:

- Capacidad para reconocer su adecuación: capacidad del producto que permite al usuario entender si el software es adecuado para sus necesidades.
- Capacidad de aprendizaje: capacidad del producto que permite al usuario aprender su aplicación.
- Capacidad para ser usado: capacidad del producto que permite al usuario operarlo y controlarlo con facilidad.
- Protección contra errores de usuario: capacidad del sistema para proteger a los usuarios de hacer errores.
- Estética de la interfaz de usuario: capacidad de la interfaz de usuario de agrandar y satisfacer la interacción con el usuario.
- Accesibilidad: capacidad del producto que permite que sea utilizado por usuarios con determinadas características y discapacidades.

## 5. Fiabilidad

Capacidad de un sistema o componente para desempeñar las funciones especificadas, cuando se usa bajo unas condiciones y periodo de tiempo determinados. Esta característica se subdivide a su vez en las siguientes subcaracterísticas:

- Madurez: capacidad del sistema para satisfacer las necesidades de fiabilidad en condiciones normales.
- Disponibilidad: capacidad del sistema o componente de estar operativo y accesible para su uso cuando se requiere.

- Tolerancia a fallos: capacidad del sistema o componente para operar según lo previsto en presencia de fallos hardware o software.
- Capacidad de recuperación: capacidad del producto software para recuperar los datos directamente afectados y reestablecer el estado deseado del sistema en caso de interrupción o fallo.

## 6. Seguridad

Capacidad de protección de la información y los datos de manera que personas o sistemas no autorizados no puedan leerlos o modificarlos. Esta característica se subdivide a su vez en las siguientes subcaracterísticas:

- Confidencialidad: capacidad de protección contra el acceso de datos e información no autorizados, ya sea accidental o deliberadamente.
- Integridad: capacidad del sistema o componente para prevenir accesos o modificaciones no autorizados a datos o programas de ordenador.
- No repudio: capacidad de demostrar las acciones o eventos que han tenido lugar, de manera que dichas acciones o eventos no puedan ser repudiados posteriormente.
- Responsabilidad: capacidad de rastrear de forma inequívoca las acciones de una entidad.
- Autenticidad: capacidad de demostrar la identidad de un sujeto o un recurso.

## 7. Mantenibilidad

Esta característica representa la capacidad del producto software para ser modificado efectiva y eficientemente, debido a necesidades evolutivas, correctivas o perfectivas. Esta característica se subdivide a su vez en las siguientes subcaracterísticas:

- Modularidad: capacidad de un sistema o programa de ordenador (compuesto de componentes discretos) que permite que un cambio en un componente tenga un impacto mínimo en los demás.
- Reusabilidad: capacidad de un activo que permite que sea utilizado en más de un sistema software o en la construcción de otros activos.
- Analizabilidad: facilidad con la que se puede evaluar el impacto de un determinado cambio sobre el resto del software, diagnosticar las deficiencias o causas de fallos en el software, o identificar las partes a modificar.
- Capacidad para ser modificado: capacidad del producto que permite que sea modificado de forma efectiva y eficiente sin introducir defectos o degradar el desempeño.
- Capacidad para ser probado: facilidad con la que se pueden establecer criterios de prueba para un sistema o componente y con la que se pueden llevar a cabo las pruebas para determinar si se cumplen dichos criterios.

## 8. Portabilidad

Capacidad del producto o componente de ser transferido de forma efectiva y eficiente de un entorno hardware, software, operacional o de utilización a otro. Esta característica se subdivide a su vez en las siguientes subcaracterísticas:

- Adaptabilidad: capacidad del producto que le permite ser adaptado de forma efectiva y eficiente a diferentes entornos determinados de hardware, software, operacionales o de uso.
- Capacidad para ser instalado: facilidad con la que el producto se puede instalar y/o desinstalar de forma exitosa en un determinado entorno.

- Capacidad para ser reemplazado: capacidad del producto para ser utilizado en lugar de otro producto software determinado con el mismo propósito y en el mismo entorno.

La norma Internacional ISO/IEC 25010 está derivada de ISO/IEC 9126:1991, *Ingeniería de Software – Calidad de producto*. Esta definía seis características de calidad y describía un modelo del proceso de evaluación del producto de software que a su vez fue reemplazada por dos normas multi-partes relacionadas: ISO/IEC 9126, *Ingeniería de Software – Calidad del producto* e ISO/IEC 14598, *Ingeniería de Software – Evaluación del producto*. La norma ISO/IEC 25010 revisa ISO/IEC 9126-1:2001, e incorpora las mismas características de calidad de producto con algunas enmiendas:

- El objeto y campo de aplicación de los modelos de calidad ha sido extendido para incluir los sistemas de computador y la calidad en uso desde la perspectiva del sistema.
- Cobertura de contexto ha sido adicionada como una característica de calidad en uso, con las sub-características integridad de contexto y flexibilidad.
- *Seguridad* ha sido adicionada como una característica, más que como una sub-característica de una funcionalidad, con sub-características confidencialidad, integridad, no-repudio, responsabilidad y autenticidad.
- Compatibilidad (incluyendo interoperabilidad y co-existencia) ha sido adicionada como característica.
- Las siguientes sub-características han sido adicionadas: integridad funcional, capacidad, protección de error de uso, accesibilidad, disponibilidad, modularidad y reusabilidad.
- Las sub-características de cumplimiento han sido retiradas, ya que el cumplimiento con las leyes y regulaciones es parte de los requerimientos del sistema general, más que específicamente parte de la calidad.
- Los modelos de calidad internos y externos han sido combinados como modelo de calidad del producto.
- Cuando es apropiado, las definiciones genéricas han sido adoptadas, más que el uso de definiciones específicas de software.
- A varias características y sub-características se les ha dado nombres más precisos. [F].

#### 2.3.4.2.- ISO/IEC 25022:2016, *Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Measurement of quality in use*

La norma ISO / IEC 25022: 2016 define las medidas de calidad en uso para las características definidas en la norma ISO/IEC 25010 y está diseñada para ser utilizada juntas.

No asigna rangos de valores de las medidas a niveles nominales o a grados de cumplimiento porque estos valores se definen para cada sistema o producto dependiendo del contexto de uso y de las necesidades de los usuarios.

Los principales usuarios de ISO/IEC 25022: 2016 son personas que llevan a cabo actividades de especificación y evaluación como parte de lo siguiente:

- Desarrollo: incluye análisis de requisitos, diseño y pruebas mediante aceptación durante el proceso del ciclo de vida.
- Gestión de la calidad: examen sistemático del producto o del sistema informático, por ejemplo, al evaluar la calidad en el uso como parte de la garantía de calidad y el control de calidad.

- Suministro: un contrato con el adquiriente para el suministro de un sistema, producto de software o servicio de software en virtud de un contrato, por ejemplo, al validar la calidad en la prueba de calificación.
- Adquisición: incluye la selección de productos y las pruebas de aceptación, al adquirir o adquirir un sistema, producto de software o servicio de software de un proveedor.
- Mantenimiento: mejora del producto en función de las medidas de calidad en el uso. [H].

#### 2.3.4.3.- ISO/IEC 25023:2016, *Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Measurement of system and software product quality*

Define las medidas de calidad para evaluar cuantitativamente la calidad del producto del sistema y del software en términos de características y subcaracterísticas definidas en la norma ISO/IEC 25010 y está diseñada para ser utilizadas juntas.

ISO/IEC 25023: 2016 no asigna rangos de valores de las medidas a los niveles nominales o a los grados de cumplimiento porque estos valores se definen en base a la naturaleza del sistema, producto o parte del producto y dependiendo de factores tales como Categoría del software, nivel de integridad y necesidades de los usuarios. Algunas características podrían tener un rango deseable de valores, que no depende de las necesidades específicas del usuario, sino que depende de factores genéricos; por ejemplo, factores cognitivos humanos.

Los principales usuarios de ISO/IEC 25023: 2016 son personas que realizan las especificaciones de requisitos de calidad y las actividades de evaluación [I].

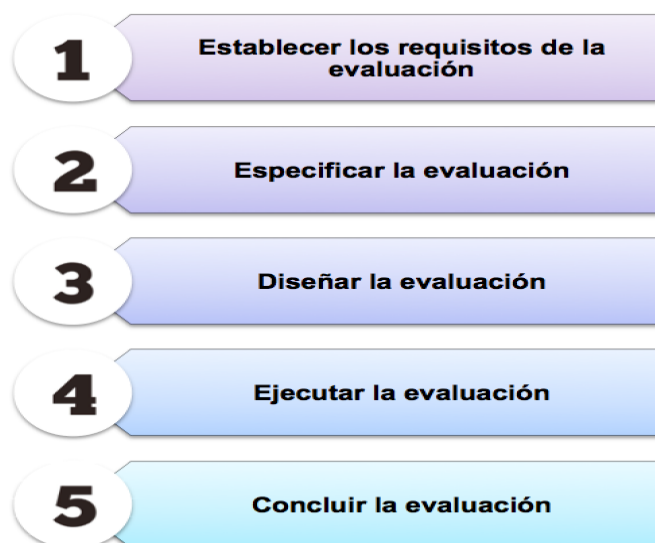
#### 2.3.4.4.- ISO/IEC 25040:2011, *Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Evaluation process*

Contiene requisitos y recomendaciones para la evaluación de la calidad del producto de software y aclara los conceptos generales. Proporciona una descripción del proceso para evaluar la calidad del producto de software y establece los requisitos para la aplicación de este proceso. Este proceso se puede usar para diferentes propósitos y enfoques, como evaluar la calidad del software pre-desarrollado, el software comercial o el software personalizado y se puede usar durante o después del proceso de desarrollo.

Está destinada a los responsables de la evaluación de productos de software y es apropiado para desarrolladores, adquirentes y evaluadores independientes de productos de software. No está destinado a la evaluación de otros aspectos de los productos de software (como los requisitos funcionales, los requisitos del proceso, los requisitos del negocio, etc.).

Esta norma define el proceso para llevar a cabo la evaluación del producto software. Dicho proceso de evaluación consta de un total de cinco actividades. [J].





**Ilustración 8 – Actividades del proceso de evaluación de calidad según ISO/IEC 25040 – Evaluation Process**

**Actividad 1:** Establecer los requisitos de la evaluación

El primer paso del proceso de evaluación consiste en establecer los requisitos de la evaluación.

Tarea 1.1: Establecer el propósito de la evaluación

En esta tarea se documenta el propósito por el que la organización quiere evaluar la calidad de su producto software (asegurar la calidad del producto, decidir si se acepta un producto, determinar la viabilidad del proyecto en desarrollo, comparar la calidad del producto con productos de la competencia, etc.).

Tarea 1.2: Obtener los requisitos de calidad del producto

En esta tarea se identifican las partes interesadas en el producto software (desarrolladores, posibles adquirientes, usuarios, proveedores, etc.) y se especifican los requisitos de calidad del producto utilizando un determinado modelo de calidad.

Tarea 1.3: Identificar las partes del producto que se deben evaluar

Se deben identificar y documentar las partes del producto software incluidas en la evaluación. El tipo de producto a evaluar (especificación de requisitos, diagramas de diseño, documentación de las pruebas, etc.) depende de la fase en el ciclo de vida en que se realiza la evaluación y del propósito de ésta.

Tarea 1.4: Definir el rigor de la evaluación

Se debe definir el rigor de la evaluación en función del propósito y el uso previsto del producto software, basándose, por ejemplo, en aspectos como el riesgo para la seguridad, el riesgo económico o el riesgo ambiental. En función del rigor se podrá establecer qué técnicas se aplican y qué resultados se esperan de la evaluación.

## **Actividad 2: Especificar la evaluación**

En esta actividad se especifican los módulos de evaluación (compuestos por las métricas, herramientas y técnicas de medición) y los criterios de decisión que se aplicarán en la evaluación.

### Tarea 2.1: Seleccionar los módulos de evaluación

En esta tarea el evaluador selecciona las métricas de calidad, técnicas y herramientas (módulos de evaluación) que cubran todos los requisitos de la evaluación. Dichas métricas deben permitir que, en función de su valor, se puedan realizar comparaciones fiables con criterios que permitan tomar decisiones. Para ello se puede tener en cuenta la Norma ISO/IEC 25020.

### Tarea 2.2: Definir los criterios de decisión para las métricas

Se deben definir los criterios de decisión para las métricas seleccionadas. Dichos criterios son umbrales numéricos que se pueden relacionar con los requisitos de calidad y posteriormente con los criterios de evaluación para decidir la calidad del producto. Estos umbrales se pueden establecer a partir de benchmarks, límites de control estadísticos, datos históricos, requisitos del cliente, etc.

### Tarea 2.3: Definir los criterios de decisión de la evaluación

Se deben definir criterios para las diferentes características evaluadas a partir de las subcaracterísticas y métricas de calidad. Estos resultados a mayor nivel de abstracción permiten realizar la valoración de la calidad del producto software de forma general.

## **Actividad 3: Diseñar la evaluación**

En esta actividad se define el plan con las actividades de evaluación que se deben realizar.

### Tarea 3.1: Planificar las actividades de la evaluación

Se deben planificar las actividades de la evaluación teniendo en cuenta la disponibilidad de los recursos, tanto humanos como materiales, que puedan ser necesarios. En la planificación se debe tener en cuenta el presupuesto, los métodos de evaluación y estándares adaptados, las herramientas de evaluación, etc.

El plan de evaluación se revisará y actualizará proporcionando información adicional según sea necesario durante el proceso de evaluación.

## **Actividad 4: Ejecutar la evaluación**

En esta actividad se ejecutan las actividades de evaluación obteniendo las métricas de calidad y aplicando los criterios de evaluación.

### Tarea 4.1: Realizar las mediciones

Se deben realizar las mediciones sobre el producto software y sus componentes para obtener los valores de las métricas seleccionadas e indicadas en el plan de evaluación. Todos los resultados obtenidos deberán ser debidamente registrados.



### Tarea 4.2: Aplicar los criterios de decisión para las métricas

Se aplican los criterios de decisión para las métricas seleccionadas sobre los valores obtenidos en la medición del producto.

### Tarea 4.3: Aplicar los criterios de decisión de la evaluación

En esta última tarea se deben aplicar los criterios de decisión a nivel de características y subcaracterísticas de calidad, produciendo como resultado la valoración del grado en que el producto software cumple los requisitos de calidad establecidos.

### **Actividad 5:** Concluir la evaluación

En esta actividad se concluye la evaluación de la calidad del producto software, realizando el informe de resultados que se entregará al cliente y revisando con éste los resultados obtenidos.

### Tarea 5.1: Revisar los resultados de la evaluación

Mediante esta tarea, el evaluador y el cliente de la evaluación (en caso de existir) realizan una revisión conjunta de los resultados obtenidos, con el objetivo de realizar una mejor interpretación de la evaluación y una mejor detección de errores.

### Tarea 5.2: Crear el informe de evaluación

Una vez revisados los resultados, se elabora el informe de evaluación, con los requisitos de la evaluación, los resultados, las limitaciones y restricciones, el personal evaluador, etc.

### Tarea 5.3: Revisar la calidad de la evaluación y obtener *feedback*

El evaluador revisará los resultados de la evaluación y la validez del proceso de evaluación, de los indicadores y de las métricas aplicadas. El *feedback* de la revisión debe servir para mejorar el proceso de evaluación de la organización y las técnicas de evaluación utilizadas.

### Tarea 5.4: Tratar los datos de la evaluación

Una vez finalizada la evaluación, el evaluador debe realizar el adecuado tratamiento con los datos y los objetos de la evaluación según lo acordado con el cliente (en caso de ser una tercera parte), devolviéndolos, archivándolos o eliminándolos según corresponda.

### 2.3.5.- Beneficios

Los beneficios de utilizar SQuaRE son:

- Representar la calidad esperada del producto de software.
- Permitir una mayor eficacia en la definición del software.
- Plantear la evaluación de productos intermedios.
- Proponer una calidad final a través de las evaluaciones intermedias.
- Mejorar la calidad del producto. [F].

### 2.3.6.- Vistas

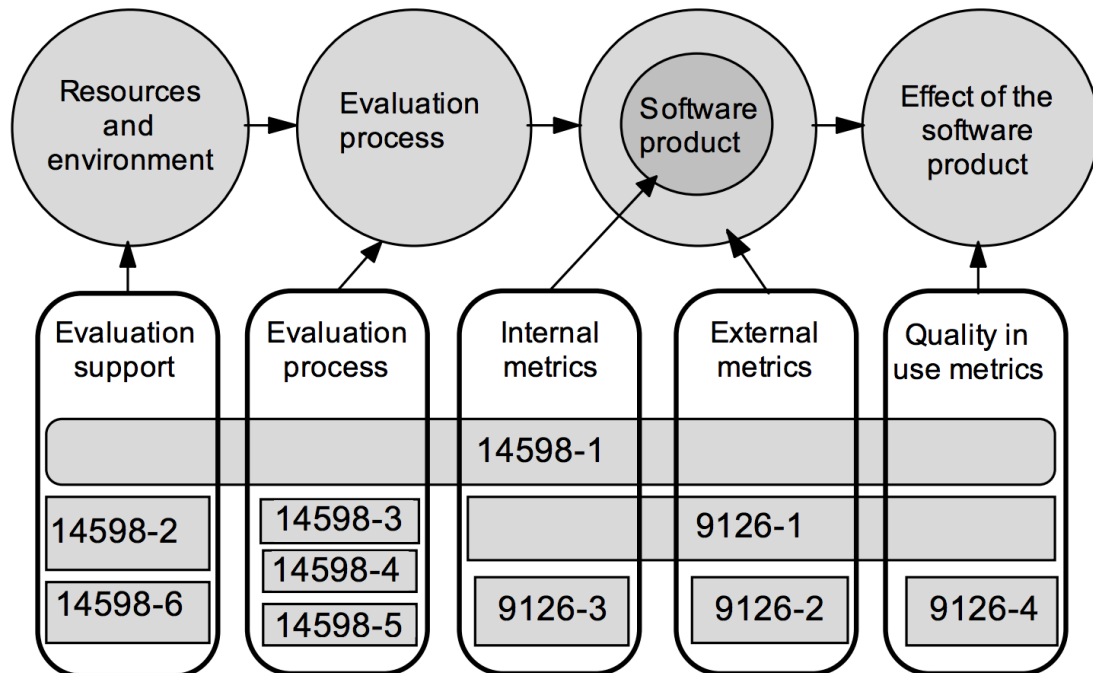
Al igual que la norma ISO/IEC 9126, la ISO/IEC 2501n – *Modelo de calidad* define tres vistas, diferenciadas en el estudio de la calidad de un producto:

- Vista interna: esta vista se ocupa de las propiedades del software como: el tamaño, la complejidad o la conformidad con las normas de orientación a objetos.
- Vista externa: vista que analiza el comportamiento del software en producción y estudia sus características, por ejemplo: el rendimiento de un software en una máquina determinada, el uso de memoria de un programa o el tiempo de funcionamiento entre fallos.
- Vista en uso: mide la productividad y efectividad del usuario final al utilizar el software.

La primera puede utilizarse desde las primeras fases del desarrollo, permitiendo detectar deficiencias en el software en edades muy tempranas del ciclo de vida del software. La segunda, sin embargo, necesita que el producto software este completo y se utiliza por tanto en el pase a producción del producto. Por último, la tercera vista que también estudia el producto de software finalizado será dependiente del usuario. La serie ISO 25000 no establece los niveles de calidad deseables para cada proyecto, si bien se recomienda que los requisitos de calidad deban ser proporcionales a las necesidades de la aplicación y lo crítico que sea el correcto funcionamiento del sistema implementado.

### 2.3.7.- Relación con normas ISO/IEC 9126 e IRAM-ISO/IEC 14598

La relación existente entre las normas ISO/IEC 9126 e IRAM-ISO/IEC 14598 se puede observar en la ilustración 9.



**Ilustración 9: Relación entre las normas ISO/IEC 9126 e IRAM-ISO/IEC 14598. [P]**

La familia de normas ISO/IEC 25000 fue explicada en puntos anteriores de este trabajo. A continuación, en la Ilustración 10, se puede observar cómo esta familia de normas se basó en la ISO/IEC 9126 e IRAM-ISO/IEC 14598 para su definición y reemplazó a estas dos normas más antiguas abarcando todos los puntos clave que estas dos últimas normas definían. En la columna izquierda se encuentran las dos últimas normas mencionadas anteriormente con sus apartados. En la columna derecha se encuentra toda la familia de normas SQuaRE. Las flechas que se observan en la columna del centro indican los apartados de la izquierda que se corresponden con los de la derecha.

CURRENT		SQuaRE
<b>9126: Product quality</b>		<b>25000: Quality Management Division</b>
-1: Quality model		25000: Guide to SQuaRE (NP)
-2: External metrics		25001: Planning and management
-3: Internal metrics		<b>25010: Quality Model Division</b>
-4: Quality in use metrics		25010: Quality model (Rev)
		<b>25020: Quality Measurement Division</b>
<b>New Proposal</b>		25020: Measurement reference model and guide (NP)
Guides to use 9126 & 14598		25021: Measurement primitives (NP)
Base metrics		25022: Measurement of internal quality
Quality requirements		25023: Measurement of external quality
		25024: Measurement of quality in use
<b>14598: Product evaluation</b>		<b>25030: Quality Requirements Division</b>
-1: General overview		25030: Quality requirements (NP)
-2: Planning and management		<b>25040: Quality Evaluation Division</b>
-3: Proc for developers		25040: Quality evaluation reference model and guide
-4: Proc for acquirers		25041: Evaluation modules
-5: Proc for evaluators		25042: Process for developers
-6: Doc of evaluation modules		25043: Process for acquirers
		25044: Process for evaluators

**Ilustración 10: Relación y correspondencia entre cada parte de la familia de las normas ISO/IEC 25000 con las normas ISO/IEC 9126 e IRAM-ISO/IEC 14598. [P]**

Las siguientes partes de la norma representan un agregado respecto de sus antecesoras:

- ISO/IEC 25000 – Guía de SQuaRE: Guía completa de toda la norma.
- ISO/IEC 25012 – Modelo de Calidad de los Datos
- ISO/IEC 25020 – Guía y Modelo de Referencia de Mediciones
- ISO/IEC 25021 – Elementos de Métricas de Calidad
- ISO/IEC 25024 – Métricas de Calidad de los Datos
- ISO/IEC 25030 – Requerimientos de Calidad
- ISO/IEC 25045 – Módulo de Evaluación para la Recuperabilidad
- ISO/IEC 25050 a ISO/IEC 25099: Números de normas reservadas para extender la norma actual.

Se observa que SQuaRE presenta una mejor arquitectura de las normas que la componen, clasificándolas por dominios o apartados relacionados a la calidad y ofrece una guía general de la norma y facilitar su comprensión y navegabilidad.

En cuanto a las métricas incorpora elementos para la medición de calidad de los datos, y un modelo de referencia común a todas ellas, incluyendo una guía para la selección y aplicación de métricas ISO y para el desarrollo de nuevas métricas. Además sugiere un conjunto de recomendado de métricas básicas y derivadas.

## 2.4.- Proceso de mejora continua

La Gestión de Calidad del Software, aplicada a una organización es el conjunto de actividades que dicha organización debe definir para lograr los objetivos de calidad de sus productos, las responsabilidades que conllevan los mismos, y la administración de los recursos para lograrlos.

La mejora continua, debe ser el objetivo permanente. Para ello se utiliza un ciclo PDCA, el cual se basa en el principio de mejora continua de la gestión de la calidad.

La base de la mejora continua es la autoevaluación. En ella se detectan puntos fuertes, que hay que tratar de mantener y áreas de mejora.

La mejora continua en los procesos y productos de software es lo que se conoce como calidad total, que en gran parte es conseguida a través de la aplicación y evaluación de las métricas, y la realimentación del proceso productivo con estos resultados [D].

El ciclo PDCA, también conocido como "Círculo de Deming", de mejora continua constituye una estrategia de mejora continua de la calidad en cuatro pasos (véase ilustración 11):



**Ilustración 11 – Proceso de mejora continua**

El ciclo, sin entrar en detalles, se interpreta de la siguiente manera: cuando se busca obtener algo, lo primero que hay que hacer es planificar cómo conseguirlo, después se procede a realizar las acciones planificadas, a continuación se comprueba qué tal se ha hecho y finalmente se implementan los cambios pertinentes para no volver a incurrir en los mismos errores. Nuevamente se empieza el ciclo planificando su ejecución pero introduciendo las mejoras provenientes de la experiencia anterior. Lo aconsejable es ir consolidando los ciclos de mejoras a través del uso de estándares para todas las actividades del mismo.

#### 2.4.1.- Proceso de mejora continua aplicado a la familia de normas ISO/IEC 25000

En este trabajo se evaluará la calidad de una aplicación móvil aplicando la familia de normas ISO/IEC 25000, tomando el modelo y las métricas a utilizar de la norma ISO/IEC 25010-*System and software quality models* y las divisiones de la evaluación de la calidad de la norma ISO/IEC 25040-Evaluation process para el proceso de evaluación de la aplicación móvil particular, en el marco de un proceso de mejora continua.

En la tabla 7 se puede observar la integración del proceso de mejora continua con la evaluación de la calidad del producto de software.

Pasos del ciclo de Mejora Continua	Relación con este trabajo
Verificar (Check)	En ésta etapa del proceso de mejora continua se realiza la evaluación de calidad del producto de software en cuestión, comprobando resultados. Se aplica la familia de las normas ISO/IEC 25000, y para la evaluación del producto de software se utiliza en particular, la norma ISO/IEC 25040. El proceso de evaluación comienza a partir de los requerimientos de la evaluación (capítulo 5) y finaliza con el reporte de la evaluación (capítulo 9).
Actuar (Act)	A partir de la evaluación se obtienen resultados que pueden llegar a ser tanto positivos como negativos. De estos últimos se deben mejorar diferentes aspectos y realizar varias evaluaciones, ya que en cada una de estas se encontrarán fallas. A partir de esto se realizan diversas acciones a seguir, como por ejemplo, tareas de mantenimiento, capacitación del personal, división de grupos de trabajo y procesos, tareas de mantenimiento, adopción de nuevos estándares, etc. La etapa de Actuar, donde se definen las acciones a realizar en base a la evaluación o verificación se corresponde con el capítulo 9, sección de acciones de mejora propuestas.
Planificar (Plan)	Este paso consiste en planificar a lo largo de un período definido las acciones especificadas en la etapa de Actuar del proceso de mejora continua.
Hacer (Do)	Etapa en la cual se realizan las acciones de mejora planificadas con anterioridad.

Tabla 7 - Relación entre el ciclo PDCA y este trabajo

El presente trabajo persigue el objetivo de alimentar el proceso de mejora para poder ser continuado en trabajos futuros, teniendo en cuenta, además, que las etapas de "Planificar" y "Hacer" comienzan luego de las conclusiones o resultados obtenidos a partir del paso "Actuar" y perduran por un tiempo establecido a priori, haciendo que este trabajo se exceda de los objetivos principales predeterminados.

## 2.5.- Conclusiones

En este capítulo se mostraron diferentes definiciones de calidad en el software,. También se evidenció la importancia que tiene la calidad del software tanto para la organización desarrolladora como los usuarios.

Luego se fijaron ciertos conceptos relacionados con la medición de la calidad en un producto de software y se realizó un estudio de la familia de las normas ISO/IEC 25000 (SQuaRE), abarcando la descripción de cada una de las sub-normas dentro de la ISO 25000 (ISO/IEC 25010-, la ISO/IEC 25022-, la ISO/IEC 25023- e ISO/IEC 25040-). Además, se muestran los beneficios de utilizar esta familia de normas para realizar la evaluación de calidad y las diferentes vistas que posee, relacionando y explicando también, cuáles son las ventajas que posee sobre las normas ISO/IEC-9126 - *Calidad del Producto* e IRAM-ISO/IEC 14598 - *Evaluación del Producto de*

*Software.* Por último, se fijó el concepto de "proceso de mejora continua", mostrando el ciclo de Deming PDCA (Plan, Do, Check, Act) y siguiendo cada uno de los casos relacionándolo con la evaluación de calidad sobre una aplicación móvil que se realizará a partir del capítulo 5 de este trabajo, llegando a la conclusión de que este proceso puede llevarse a cabo por todas las organizaciones sin importar su tamaño o tipo.



---

## CAPITULO 3.- CALIDAD EN APLICACIONES MOVILES

---

### 3.1.- Aplicación móvil

Las tecnologías móviles y su continuo avance están propiciando una nueva generación de aplicaciones, estas son las denominadas “aplicaciones móviles”. Se considera aplicación móvil, a aquel software desarrollado para dispositivos móviles. Móvil se refiere a poder acceder desde cualquier lugar y momento a los datos, las aplicaciones y los dispositivos. Este tipo de aplicaciones se desarrollan teniendo en cuenta las limitaciones de los propios dispositivos, como por ejemplo el bajo poder de cómputo, la escasa capacidad de almacenamiento, ancho de banda limitado, etc. Los dispositivos móviles son suficientemente livianos como para ser transportados por personas y disponen de la capacidad de batería adecuada para funcionar de forma autónoma. Algunos de estos dispositivos se pueden observar en la ilustración 12.

Estos dispositivos están dominados por diferentes plataformas tecnológicas, incluyendo diferentes sistemas operativos. Cada uno tiene sus particularidades en cuanto al manejo por parte del usuario, como así también al momento de desarrollar una aplicación. Los sistemas operativos para móviles son mucho más simples que los de una computadora y están más orientados a la conectividad inalámbrica.



**Ilustración 12 – Ejemplos de dispositivos móviles**

Existen dos categorías en las que se pueden clasificar las aplicaciones móviles: aplicaciones nativas y aplicaciones Web.

1.Aplicaciones nativas: Las aplicaciones nativas son desarrolladas específicamente para un tipo de dispositivo y su sistema operativo, se basan en la instalación de código ejecutable en el dispositivo del usuario. Estas tienen la ventaja de acceder a las funciones del dispositivo, como por ejemplo: almacenamiento, GPS (sistema de posicionamiento global), SMS (servicio de mensajes cortos), mails, etc. Existen repositorios de los cuales se pueden descargar e instalar este tipo de aplicaciones, según el sistema operativo. El principal inconveniente de estas aplicaciones es que se



deben desarrollar para cada plataforma y por lo tanto incrementa el tiempo de desarrollo, costo y esfuerzo.

**2Aplicaciones Web:** Las aplicaciones móviles de este tipo se encuentran ejecutándose en servidores, estas incluyen páginas web optimizadas para ser visualizadas en dispositivos móviles y se pueden desarrollar en HTML, Java Script, CSS, etc. Por definición, estas aplicaciones serán accedidas utilizando algún navegador web. La ventaja que tiene desarrollar aplicaciones móviles Web es que son fáciles de implementar y de integrar con aplicaciones existentes, además de necesitar menos requerimientos del hardware de los dispositivos móviles. El problema que tienen es que no pueden acceder a las funcionalidades propias del dispositivo. Por ejemplo, una aplicación web no puede emplear la cámara de un Smartphone, en el caso que la tuviera, para capturar imágenes o realizar una filmación.

### 3.2.- Desarrollo de aplicaciones móviles

Debido a la heterogeneidad de los dispositivos móviles se requiere que el desarrollador lleve a cabo una cuidadosa planificación y diseño antes de la implementación de la aplicación. Ejecutar una aplicación en un dispositivo móvil introduce una serie de consideraciones que un desarrollador/programador debe conocer:

- Distintas velocidades y características de la red. Al ser dispositivos móviles atravesarán distintas redes.
- Errores de red. La falta de una red de datos disponible es muy probable, en estos casos la aplicación debe tener un modo fuera de línea.
- Variación del rendimiento de la plataforma de hardware. Lo ideal es que la aplicación esté disponible para tantos dispositivos como sea posible. Esto significa soportar diferentes dispositivos y diferentes plataformas.
- Distintos tamaños y resoluciones de pantallas. Los diferentes dispositivos cuentan con diferentes pantallas con distintas funcionalidades.
- Dificultad de probar las aplicaciones por completo. Dada la variedad de dispositivos, se torna difícil probar todos los dispositivos actuales y los nuevos que ingresan al mercado.

Las aplicaciones de este tipo tienen que manejar ciertos requerimientos:

- Potencial interacción con otras aplicaciones.
- Manejo de sensores como por ejemplo el acelerómetro para responder a movimientos del dispositivo o pantalla táctil para responder a gestos.
- Seguridad en la aplicación. Numerosos dispositivos utilizan software embebido, por lo tanto son cerrados y no resulta sencillo penetrar el software incorporado. En cambio las plataformas móviles son abiertas, lo que permite instalación de software malicioso que puede afectar el funcionamiento general del dispositivo.
- Bajo consumo de energía. Muchos aspectos de una aplicación afectan el consumo de energía, con esto se ve afectada la capacidad de la batería, es decir la vida útil del dispositivo.

### 3.3.- Contexto Móvil

Los dispositivos junto con las aplicaciones móviles son utilizados en un cierto contexto, donde las características del mismo cambian continuamente. Al utilizarlos, los usuarios, tendrán diferentes objetivos, realizarán diferentes tareas; también se manipularán en diferentes entornos físicos y sociales. Todos estos factores y otros, relacionados al contexto móvil, influyen en la forma de uso de una aplicación.

Si no se tiene en cuenta el contexto como parte integral de la aplicación se estaría realizando un análisis sesgado (de características tan importantes como lo es *usabilidad*) en comparación con lo que sucede cuando un usuario usa la aplicación en el mundo real. En ese entorno real la conectividad (ancho de banda) puede ir cambiando según el lugar donde se encuentre el usuario, afectando el uso de la aplicación.

Para el dominio de las aplicaciones móviles cuyo contexto de uso cambia continuamente, existen los siguientes desafíos:

- Entorno móvil: se puede definir como cualquier información que caracteriza una situación relacionada con la interacción entre usuarios, aplicaciones y sus alrededores. Esto incluye la ubicación, las identidades de las personas cercanas, objetos, elementos del entorno que pueden distraer la atención del usuario, etc. Este puede estar cambiando continuamente.
- Conectividad: el ancho de banda limitado y la baja confiabilidad de las conexiones, son características comunes en la conectividad de estos tipos de dispositivos. Este problema afecta principalmente a los tiempos de descarga de datos y la calidad de la transmisión de audio y video. La intensidad de la señal y la velocidad de transferencia de la red pueden variar en diferentes momentos y lugares, agravadas también por la movilidad del usuario.
- Capacidad de procesamiento: el poder computacional y la capacidad de memoria de los dispositivos móviles son reducidos con respecto a dispositivos considerados de escritorio.
- Pantallas pequeñas: las limitaciones físicas propias de los dispositivos móviles, como el tamaño pequeño de las pantallas afectan la *usabilidad* de una aplicación.
- Pantallas con diferentes resoluciones: la baja resolución de la pantalla puede degradar la calidad de la información multimedia mostrada. Como resultado, diferentes resoluciones de pantalla puede causar diferentes grados de *usabilidad*.
- Método de entrada de datos: el ingreso de datos en estos dispositivos no es una tarea sencilla. Diminutos botones y etiquetas limitan la eficacia y eficiencia de los usuarios en la introducción de datos, esto reduce la velocidad de entrada y aumenta los errores. En general no es eficiente y nada amigable.

### 3.4.- Calidad sobre aplicaciones móviles

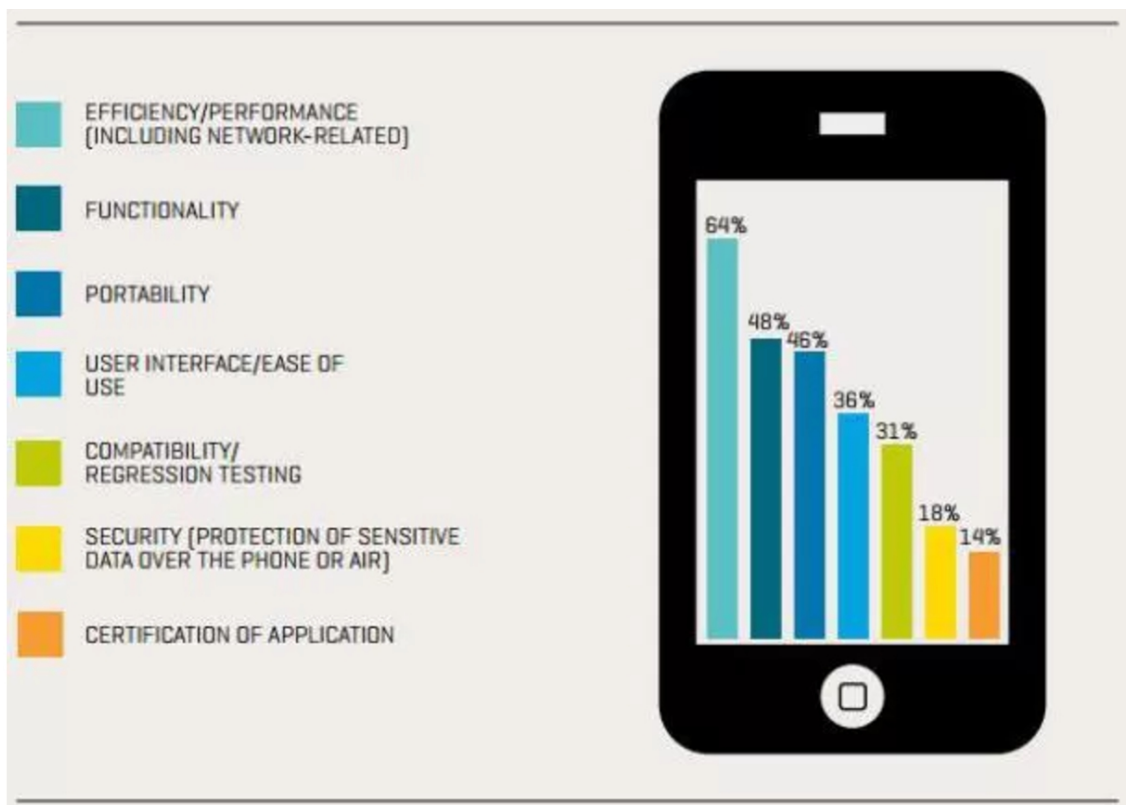
En un entorno en el que el uso de dispositivos móviles es superior al de PCs o TVs, las empresas ven a los smartphones y tablets como un canal clave donde ganar presencia, a través del cual ofrecer sus servicios o productos. Son numerosos los estudios y referencias que reflejan un cambio en la forma de relacionarse e, incluso, en el proceso de toma de decisiones del consumidor que, cada vez con mayor frecuencia, consulta información a través de su dispositivo móvil.

Dependiendo de los presupuestos y estrategias particulares, y con el objetivo de acceder al nuevo consumidor, las organizaciones optan por optimizar sus sitios web para cada dispositivo, mientras que otras se deciden por el desarrollo de aplicaciones a medida. Ahora bien, en este entorno, los usuarios son especialmente exigentes. Esperan obtener la información que necesitan de forma precisa, clara y, sobre todo, rápida. La paciencia no es una cualidad del nuevo consumidor, y en el momento en que un fallo o caída de la aplicación los retrasa o impide alcanzar su objetivo, no dudan en acudir a la competencia.

En el World Quality Report se han recopilado datos de empresas de todo el mundo que realizan controles de calidad de software y se ha identificado las principales prioridades para las pruebas con aplicaciones móviles. El rendimiento es la cualidad más buscada (por un 64% de los entrevistados) frente a la funcionalidad, que tradicionalmente más interesaba. En este reporte, llama poderosamente la atención

que menos de un 18% de las compañías encuestadas señalaron la *seguridad* como un objetivo prioritario. Esto puede significar una mayor confianza en la protección inherente de los entornos actuales [E].

La capacidad de la aplicación para funcionar en dispositivos móviles es la tercera cualidad más buscada en las empresas, siguiendo muy cerca a la funcionalidad, prioridad número uno. La certificación de aplicaciones queda como última necesidad (véase ilustración 13).

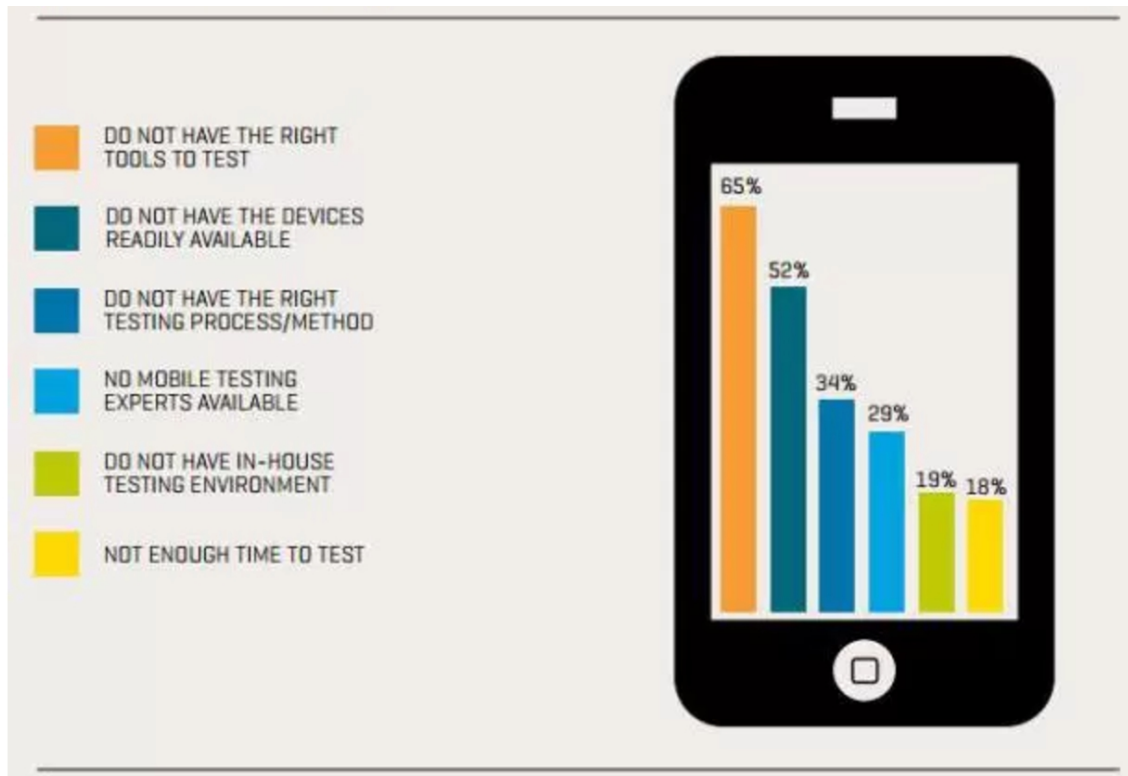


**Ilustración 13 - Características prioritarias en aplicaciones móviles**

El control de calidad en aplicaciones móviles, actualmente, es un proceso minoritario. Pero aún entre las empresas que sí realizan controles de calidad el proceso no es del todo sofisticado como debería serlo. Una gran parte de las empresas afirma que no tienen las herramientas que se requieren para hacer pruebas dirigidas a smartphones y tabletas.

También aluden a que no cuentan con los dispositivos móviles en el preciso momento en que los necesitan. Una metodología incorrecta y la inexistencia de expertos en control de calidad móvil también aparece como dificultades. Al tratarse de un campo que ha surgido y crecido con tanta rapidez, escasean los profesionales con experiencia en aplicaciones para smartphones y tabletas.

En definitiva, no se cuenta aún con los recursos necesarios, ni técnicos ni humanos, para llevar a cabo un control de calidad riguroso en aplicaciones móviles, con toda la importancia que esto conlleva.



**Ilustración 14 - Desafíos para evaluación de calidad en aplicaciones móviles de las empresas actualmente**

### 3.5.- Pruebas de software sobre aplicaciones móviles

Como en cualquier desarrollo de software la fase de testing es fundamental para cuidar la calidad del producto final y la satisfacción del cliente.

El testing en app's móviles tiene sus particularidades, ya que presenta diferencias con respecto a otros entornos y por lo tanto diferentes problemas potenciales que se pueden detectar.

Es importante tener en cuenta que existen otros tipos de pruebas de software que se realizan sobre aplicaciones móviles para mejorar la calidad de las mismas.

Los once tipos de pruebas de software más importantes sobre aplicaciones móviles son:

#### 1. Pruebas de interrupción

La ejecución de aplicaciones móviles nativas o de la web móvil puede interrumpirse por distintos eventos desencadenados por el dispositivo, como por ejemplo una llamada entrante.

Es necesario definir casos de prueba para ver el comportamiento de la aplicación ante eventos como: llamada entrante, SMS entrante, correo electrónico, notificaciones de social media, baja batería, apagado del equipo, caída de la conexión con la red, entre otros.

#### 2. Pruebas de interfaz con el usuario (UI)

Estas pruebas tienen como objetivo evaluar la interfaz con el usuario, teniendo como referencia los estándares para la interacción humano- máquina aceptable. Algunos ejemplos de este tipo de prueba son:

- Organización de pantallas, alineación, colores, fondos y patrón de lectura.
- Posición, tamaño, datos de entradas y acciones.
- Claridad, alineación y densidad de imágenes y símbolos.
- Mensajes de error.
- Consistencia de la interfaz en toda la aplicación y entre dispositivos disímiles.

### 3. Pruebas de acciones del usuario

Los móviles con pantallas táctiles han conquistado el mercado, por eso hoy en día es necesario probar cómo se comporta la aplicación ante distintas acciones como tocar, arrastrar, rotar, extender los dedos, cerrar los dedos, entre otras.

### 4. Pruebas de *usabilidad* y accesibilidad

A través de las pruebas de *usabilidad* y accesibilidad se busca validar algunos aspectos como:

- Presentación de la información en el diseño de página para móviles.
- Facilidad para completar tareas.
- Eficiencia y exactitud.
- Minimizar que el usuario tenga que recordar información.
- Tamaño de pantalla.
- Condiciones de iluminación.
- Tamaño de la interfaz táctil.
- Integración de aplicaciones nativas, aplicaciones web móvil e híbridas.

### 5. Pruebas de Movilidad

Las pruebas de movilidad prueban el desempeño de la aplicación cuando el dispositivo móvil está en movimiento, con una persona o vehículo. Son sumamente necesarias cuando se trata de aplicaciones que utilizan la localización para mostrar información relevante al usuario.

### 6. Pruebas de conectividad

Sirven para probar el desempeño de la aplicación al conectarse a las redes en distintos protocolos y distintas condiciones, por ejemplo: Wi-Fi, Bluetooth, Red analógica (3G o 4G) o señales de distinta intensidad (sin señal, conexión intermitente, señal fuerte, intensidad de señal variable).

### 7. Pruebas de *seguridad*

Su finalidad busca validar la resistencia de la aplicación a ataques maliciosos, como por ejemplo: ataques vía la red, ataques al servidor, ataques al dispositivo, entre otros.

Con este tipo de pruebas también se busca validar que los programadores apliquen prácticas de seguridad informática en el código de programa.

### 8. Pruebas de desempeño y estrés

Este tipo de prueba se realiza para determinar la solidez de la aplicación en los momentos de carga extrema y ayuda a los administradores para determinar si la

aplicación rendirá lo suficiente en caso de que la carga real supere a la carga esperada.

- Eficiencia en el consumo de batería.
- Desempeño lento.
- Cómo funciona la aplicación en modos de optimización y ahorro de energía del dispositivo.
- Pruebas con distintos niveles de batería (bajo, medio o alto).
- Utilización de procesador, memoria y espacio de almacenamiento.
- Pruebas de carga del lado de servidor (pruebas de estrés). Para lo cual pueden utilizarse herramientas como SoapUI.
- Pruebas de la red.

### 9. Pruebas de compatibilidad

Consisten en validar la compatibilidad de la aplicación con: otras aplicaciones, plataformas, dispositivos, navegadores, redes de telecomunicaciones y versiones anteriores de la aplicación.

### 10. Validación de los lineamientos para envío a las App Stores

Las aplicaciones nativas Android y iPhone solo pueden distribuirse a través de sus respectivas tiendas online (App's Stores). Las App's Store poseen una serie de lineamientos que deben cumplir las aplicaciones. Por esta razón, es una buena práctica que al testear aplicaciones móviles, se incluyan casos de prueba para validar el cumplimiento de estos lineamientos.

### 11. Pruebas de rendimiento

Tienen como objetivo ayudar a la empresa a prevenir errores que eviten el abandono de la aplicación debido a caídas o a un tiempo excesivo en su ejecución. Un alto consumo de batería, la velocidad a la que se desarrolla dependiendo del navegador utilizado o las características de cada sistema operativo, pueden ser factores que influyen de forma decisiva en este apartado.

## 3.6.- Conclusiones

El término "app" (aplicación móvil) se ha puesto muy de moda en los últimos años, y más aún con la proliferación de los dispositivos móviles, los cuales son más y superiores. Además, cada vez son más las PyMEs o empresas que poseen su aplicación móvil, o desarrollan y mantienen app's. El uso de este tipo de producto de software viene creciendo de manera exponencial, y, debido a esto, es de sumo interés en este trabajo realizar una evaluación de calidad a una aplicación móvil, teniendo en cuenta que diferentes características de calidad (*usabilidad*, etc) han comenzado a ser consideradas recientemente sobre un contexto móvil.

Se puede decir que el mercado del software móvil ha progresado más deprisa que la capacidad que tienen las empresas para adaptarse. La expansión de smartphones, tablets y otros dispositivos móviles ha sido muy significativa en los últimos años. De hecho, hoy en día no se cuenta con los recursos técnicos y la experiencia profesional para llevar a cabo una optimización adecuada en este tipo de desarrollo. Pero las bases para realizar el control de calidad para aplicaciones móviles definitivamente ya existen.

Desde el inicio de su construcción o desarrollo es necesario probar las aplicaciones móviles teniendo en cuenta, precisamente, su propio contexto de

movilidad y considerando el cumplimiento de las características de funcionalidad, *usabilidad*, *seguridad* y rendimiento (entre otras) de las mismas. La complejidad del aseguramiento de la calidad del software en aplicaciones móviles radica en la existencia de diferentes sistemas operativos y multitud de modelos de dispositivos. De esta forma, una aplicación debidamente construida y de calidad será aquella que funcione correctamente en todos los dispositivos móviles y bajo todos los sistemas operativos existentes en el mercado.

Finalmente, dedicar especial atención a las pruebas de software desde las primeras fases de su desarrollo ayudará a las empresas a posicionar sus aplicaciones con una mayor valoración, que otorga la máxima calidad en los market/stores de aplicaciones móviles, posicionándose de una mejor manera frente al mercado del software.



## CAPITULO 4.- DESCRIPCION DEL PRODUCTO A EVALUAR

---

### 4.1.- Descripción general del producto de software

#### 4.1.1.- Introducción

El producto de software que fue seleccionado para realizar la evaluación de calidad se llama *CargoX*. Es una aplicación móvil desarrollada para dispositivos móviles con sistema operativo Android que se utiliza para que los transportistas puedan visualizar cargas, añadidas por personas o empresas, y seleccionar aquellas que deseen transportar de una ciudad origen a una destino. Anteriormente existía otra aplicación móvil denominada *SontraCargo* (que hoy en día está dejándose de utilizar) desarrollada por la misma empresa que creó *CargoX*, más básica y menos intuitiva.

La organización propietaria de la aplicación móvil *CargoX* se llama *Sontra*, la cual posee su sitio web [L] y está ubicada en Brasil, donde se encuentra un mercado muy grande de camioneros y transporte de cargas. Más adelante se describirá con más detalles la información pertinente a esta empresa.

#### 4.1.2.- Dominio y funcionamiento del producto

El dominio del producto es la logística, más específicamente, todo lo que se refiere a transporte de cargas (de todo tipo) de un punto origen a un destino por parte de transportistas usuarios de la aplicación móvil *CargoX*.

Esta aplicación contiene pedidos de los clientes que pueden ser personas particulares o empresas.

Las personas o empresas publican las cargas que desean transportar (estas pueden ser de cualquier tamaño), junto con información importante como origen y destino del transporte, y datos adicionales como nombre, apellido, teléfono, etc. Esto se realiza desde un panel web o dashboard que se explicará más adelante. Una vez publicado, los pedidos pueden ser visualizados por transportistas que deciden cuál transportar.

Luego de que el chofer encargado del transporte de la carga elija el pedido, se inicia un flujo de seguimiento del transporte para que se pueda realizar un seguimiento, hasta que finalmente llega al destino y se entrega el producto.

Para fijar el precio al transporte de la carga, se utiliza una API denominada KMM (de origen brasilera) donde se obtienen precios a partir de los kilómetros que se van a recorrer, las rutas que se van a tomar, los peajes en el camino, etc.

Una vez concluido el viaje, el pedido se almacena en el historial de pedidos terminados para constatar que fue entregado correctamente, caso contrario, se comunica a la persona que no se pudo entregar el producto justificando el porque.

#### 4.1.3.- Alcance del producto

Como se mencionó con anterioridad, *CargoX* es una aplicación móvil que permite transportar cargas o fletes de un sitio origen a otro destino. Para definir el alcance hay que tener en cuenta las siguientes funciones que permite realizar la app:

- *CargoX* cuenta con un único rol, el de transportista, que utilizará sus funcionalidades (aceptar pedidos, contactar persona que desea transportar, rechazar pedido, etc).
- La aplicación permite listar las cargas pendientes para que el transportista seleccione y así iniciar el transporte.
- Permite mostrar detalles de las diferentes cargas (mapa, fecha/hora de recogida, producto, peso, lugar y fecha de entrega).
- *CargoX* muestra una pantalla con la carga actual y su respectivo estado, el cual puede ser: esperando recogida, transportando, llevando a destino, etc. Existe la posibilidad de contactar a la persona que ofrece el pedido, e informarle el estado de la misma.
- La app permite visualizar el perfil del transportista, con sus respectivos datos (nombre, apellido, teléfono, etc) para que la persona que publicó la carga a transportar pueda comunicarse con el transportista que aceptó ese cargamento.
- Permite visualizar el listado de cargas ya transportadas a través de la pantalla de "histórico de cargas" sirviendo, también, como control para el transportista.

Además de la aplicación móvil, existe un panel web, denominado dashboard, que utilizan los administradores, operadores, personas que realizan pedidos, de soporte, finanzas, seguros/legales, sistemas, etc, para poder realizar cada uno su tarea. Este dashboard se encuentra en el sitio web de Sontra [L] y para utilizarlo hay que iniciar sesión en el mismo. Por ejemplo, los administradores u operadores pueden visualizar y administrar:

- Personas interesadas en transportar alguna carga.
- Clientes.
- Camiones o transportes.
- Transportistas.
- Tracking de camiones.
- Cargas.
- Seguro de cargas (se exige que cada carga tenga seguro para poder transportarse, junto con los documentos formales que se requieren para poder realizar el transporte).

Además, las personas o empresas que desean transportar cargamento pueden realizar sus pedidos, pudiendo:

- Ver listado de cargas que desean transportar.
- Ver tracking de las cargas.
- Ver detalles del transportista.
- Publicar pedido.

El resto de los roles del dashboard se encargan de realizar tareas específicas, por ejemplo, en seguridad pueden ver las cargas para poder llevar un control, los de legales pueden constatar que todo esté dentro del marco legal y permitido, etc.

#### 4.1.4.- Objetivos

- Objetivo general: ser un nexo entre transportistas y cargas a transportar.
- Objetivos específicos:
  - Visualizar de manera más intuitiva y ágil los pedidos de cargas a transportar junto con la ciudad origen y destino para su selección por parte de los transportistas.
  - Lograr que las personas puedan desplazar sus cargas/fletes conectándose con los transportistas contando con un sistema de soporte, ayuda y seguimiento de las mismas
  - Acelerar el proceso de envío y recepción de cargamentos.
  - Brindar mayor comodidad y facilidad para los usuarios.

#### 4.1.5.- Usuarios del producto

La aplicación móvil *CargoX* va dirigida a aquellas personas que son choferes en general (de camiones en su gran mayoría) encargados de transportar cargas de una ciudad origen a una ciudad destino.

#### 4.1.6.- Arquitectura

*CargoX* corre sobre dispositivos móviles (smartphones, tablets, etc) con sistema operativo Android con versión mayor a 4.0. Para su desarrollo se utiliza el lenguaje de programación PHP con MySQL como motor de base de datos, máquinas con sistema operativo Linux con servidores en amazon para almacenamiento. Se utilizan muchas librerías de diversos tipos, como por ejemplo para la utilización del GPS para los mapas, para animaciones, reportes, etc, y Google Analytics<sup>3</sup>. La arquitectura de *CargoX* es una arquitectura de 3 capas como se ve en la ilustración 15.

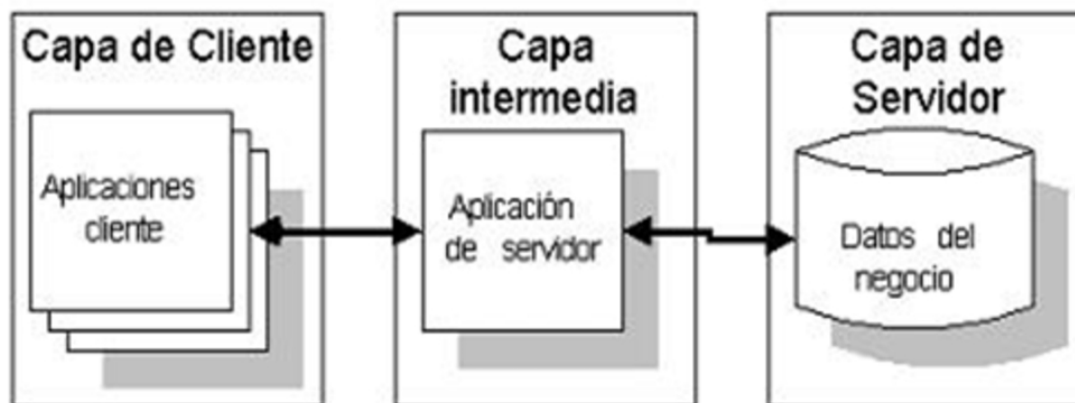


Ilustración 15 - Arquitectura del producto de software a evaluar

---

<sup>3</sup> Google **Analytics** es una herramienta de analítica web de la empresa Google. Ofrece información agrupada del tráfico que llega a los sitios web según la audiencia, la adquisición, el comportamiento y las conversiones que se llevan a cabo en el sitio web.

## 4.2.- Utilidad del producto en el mercado y para su evaluación de calidad

Para un producto de software cuyo dominio es la logística o transporte de mercaderías es muy importante tener en cuenta la calidad, ya que los riesgos que pueden presentarse pueden ser desfavorables tanto para el cliente que deposita su confianza en una aplicación para enviar su preciada carga como para los transportistas. Además, un buen nivel de calidad es necesario para mantener y aumentar la seguridad en el producto por parte de los usuarios hacia la aplicación, de modo que pueda mantenerse y crecer en el mercado. En este contexto, es importante incorporar el hábito de la realización de evaluaciones de calidad formales y estandarizadas para la mejora continua de este.

La característica *usabilidad*, es una de las seleccionadas para ser evaluadas para alcanzar una aplicación más sencilla e intuitiva para los usuarios; y *seguridad*, para evitar ingresos fraudulentos, robo de datos, etc. y así poseer una app más segura y confiable, aprovechando todo el potencial que la misma podría brindar si su nivel de calidad es mayor.

## 4.3.- Descripción detallada del producto de software

### 4.3.1.- Elementos propios del producto de software

Listado de cargas pendientes: lista de cargas o fletes pendientes de selección por parte de los transportistas. Estos seleccionan el cargamento que deseen y luego se inicia el proceso de envío.

Detalle de la carga: pantalla que muestra el detalle de la información de la carga seleccionada. Indica: el mapa, la fecha y hora de recogida, el producto a transportar, el peso, el origen, el destino y la fecha de entrega.

Carga actual: pantalla que muestra el estado de la carga (esperando recogida, transportando, llevando a destino, etc) con posibilidad de contactar a la persona que realizó el pedido.

Historial de cargas: listado de las cargas que ya fueron transportadas por parte del transportista. Sirve como control.

Perfil de usuario: Pantalla que contiene datos del transportista. Pueden ser modificados por sí mismo.

Como ya se mencionó, también existe un panel web que no se tendrán en cuenta para realizar la evaluación de calidad, ya que sólo nos enfocaremos en la app.

### 4.3.2.- Pantallas del producto de software

A continuación se muestra por medio de ilustraciones la aplicación móvil CargoX:

Login y registro (12 pantallas):

Las primeras cuatro pantallas que se observan en la ilustración 16 muestran algunos de los datos a ingresar y mensajes para el login de CargoX:

- La primera pantalla solicita al usuario el número de su celular.
- La segunda pantalla muestra el ingreso del número del celular del usuario.
  - Si el número existe, se deriva directamente a la pantalla de ingreso de CPF, explicado más adelante.
  - Si no existe, se deriva al ingreso del código SMS enviado al número de celular, explicado en el siguiente punto.

- La tercer pantalla muestra los campos para ingresar el código SMS enviado al número de celular del usuario.
- La cuarta pantalla muestra el código SMS ingresado por el usuario.



**Ilustración 16 – Pantallas del login y registro**

Las siguientes cuatro pantallas que se observan en la ilustración 17 muestran más datos a ingresar y mensajes para el login de CargoX:

- La primer pantalla muestra que se está reenviando el código SMS (si este nunca le llegó al usuario).
- La segunda pantalla muestra que ya se envió el código SMS al número de celular ingresado previamente.
- La tercer pantalla muestra al usuario que no existe un CPF<sup>4</sup>, ingresado por el mismo (véase en la ilustración 18 el ingreso de CPF), en la aplicación y solicita ciertos datos.
- La cuarta pantalla es igual a la tercer pantalla pero con los datos completados por el usuario.



**Ilustración 17 – Pantallas del login y registro (cont. 1)**

Las siguientes cuatro pantallas que se observan en la ilustración 18 muestran mensajes de error o de éxito del inicio de sesión y datos a ingresar para la funcionalidad de login de CargoX:

- La primer pantalla muestra que se ingresó un CPF correcto y ya se puede ingresar a la app.

---

<sup>4</sup>El CPF (Cadastro de Pessoas Físicas or Natural Persons Register) es un **número** asignado por la agencia de ingresos de Brasil tanto a brasileños como a extranjeros residentes que pagan impuestos en Brasil.

- La segunda pantalla muestra que el número de celular fue confirmado correctamente y existe en el sistema y solicita número de CPF.
- La tercer pantalla muestra el ingreso del número CPF por parte del usuario.
- La cuarta pantalla informa al usuario que el número CPF no es correcto o no existe.



**Ilustración 18 – Pantallas del login y registro (cont. 2)**

Cargas (2 pantallas):

Las siguientes dos pantallas que se observan en la ilustración 19 muestran todo lo relacionado con las cargas de CargoX:

- La primer pantalla muestra los pedidos de transporte de cargas indicando para cada uno la fecha y hora a recoger la carga, el lugar donde se recoge y hacia donde se transporta.
- La segunda pantalla muestra el recorrido del transporte de la carga seleccionada una vez que se inicia el mismo, junto con toda la información de la carga y el transporte.

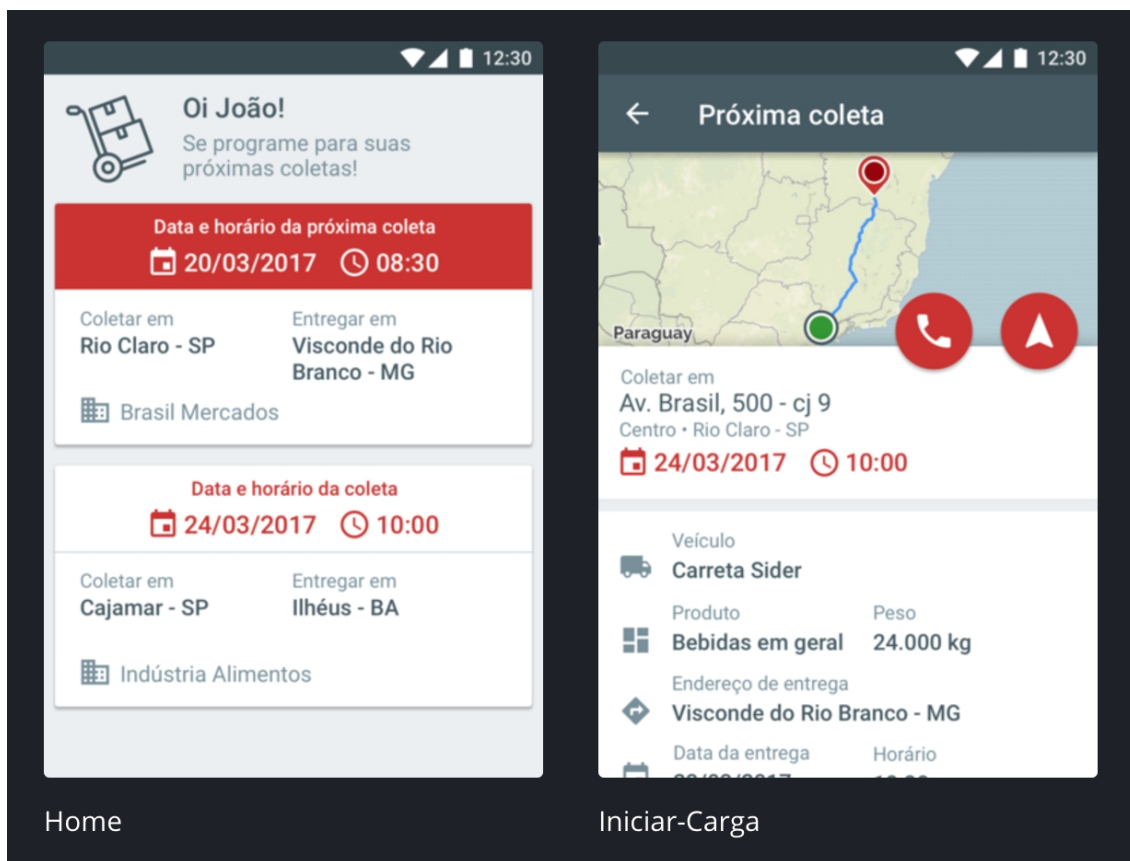


Ilustración 19 – Pantalla de cargas

En ruta (6 pantallas):

Las siguientes seis pantallas que se observan en las ilustraciones 20 y 21 muestran todo lo relacionado con el recorrido de las cargas de una vez iniciado el transporte dentro de CargoX. A medida que el transportista va avanzando en el recorrido la app muestra los diferentes cambios de estado (yendo a coleccionar, aguardando para cargar, cargando el camión, en tránsito, aguardando para descargar, descargando).

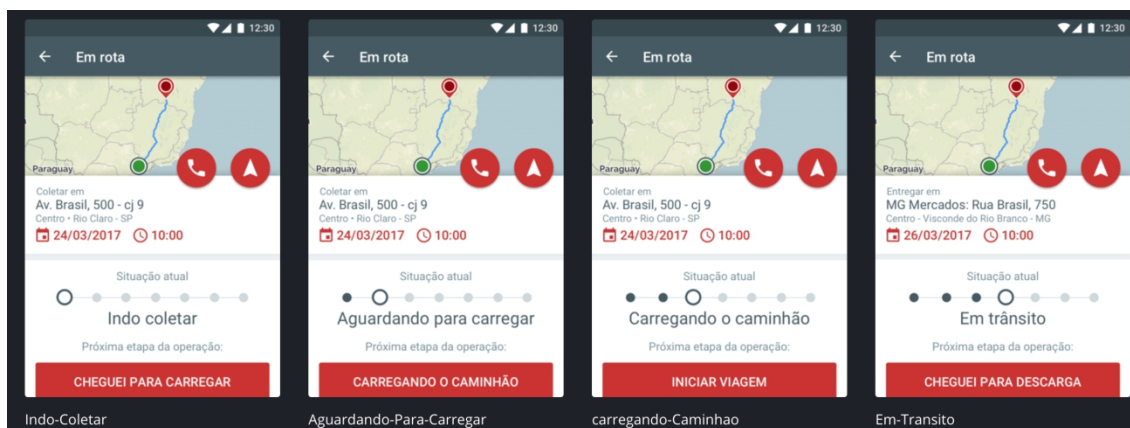
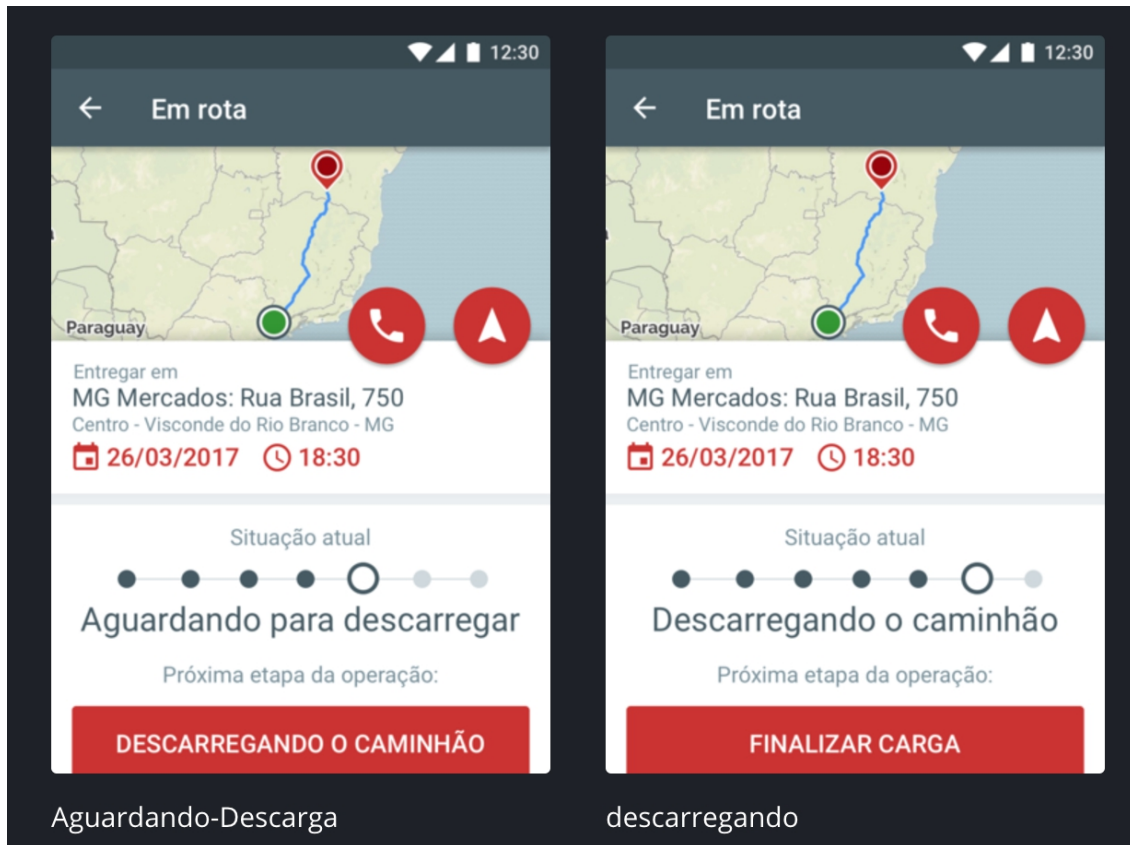


Ilustración 20 – Pantallas de recorrido del cargamento





**Ilustración 21 – Pantallas del recorrido del cargamento (cont.)**

Historial de carga (2 pantallas):

Las siguientes dos pantallas que se observan en la ilustración 22 muestran todo lo relacionado con el historial de cargas de CargoX:

- La primera pantalla muestra las cargas que ya terminaron, mostrando para cada una la fecha, hora, lugar de recogida, lugar de entrega y si fue entregada correctamente.
- La segunda pantalla muestra detalles de cada transporte ya finalizado al seleccionar cada uno de ellos en la pantalla explicada anteriormente.

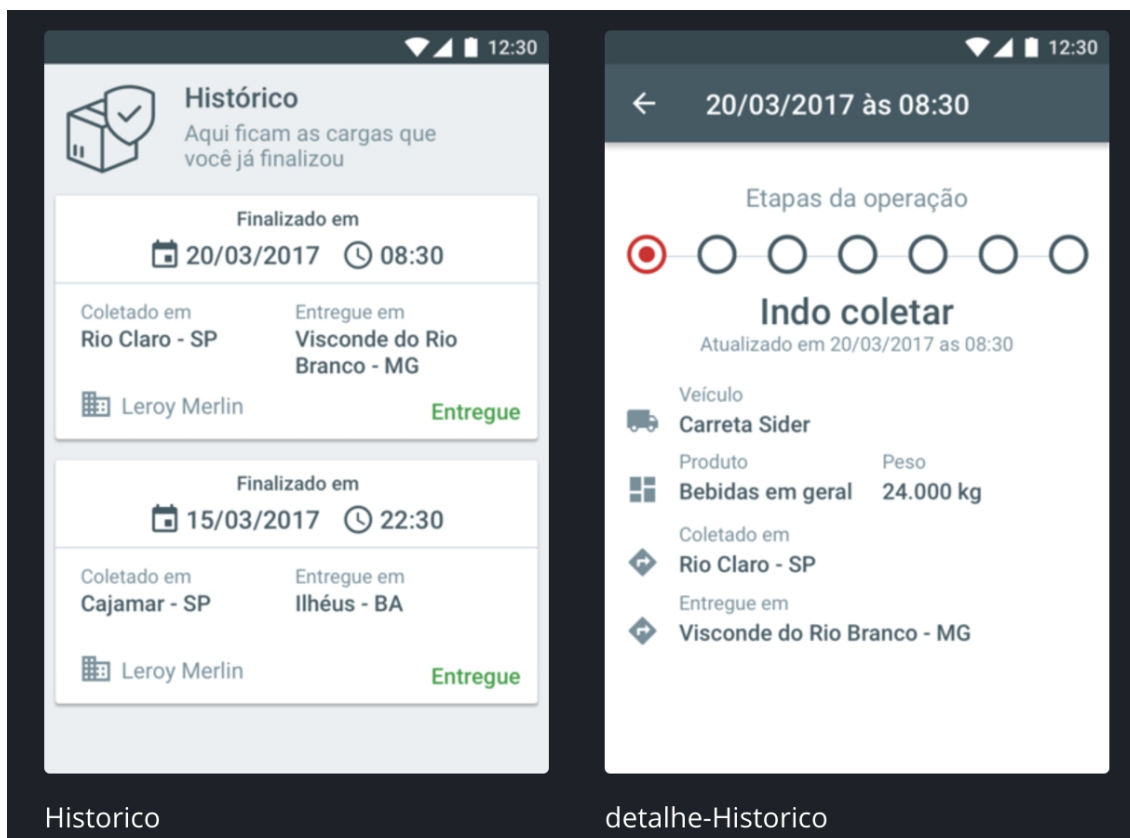


Ilustración 22 – Pantallas de historial de cargas

Perfil de usuario (7 pantallas):

Las siguientes cuatro pantallas que se observan en la ilustración 23 muestran el perfil del usuario dentro de la aplicación móvil CargoX:

- La primer pantalla muestra el menú del perfil del usuario transportista (datos de la cuenta, rutas favoritas, ayuda y salir).
- La segunda pantalla muestra datos importantes de la cuenta del usuario.
- La tercera pantalla muestra las rutas favoritas que fueron agregadas por el usuario.
- La cuarta, quinta, sexta y séptima pantalla (véase en ilustración 24) muestran como agregar una ruta favorita y la sección de ayuda.

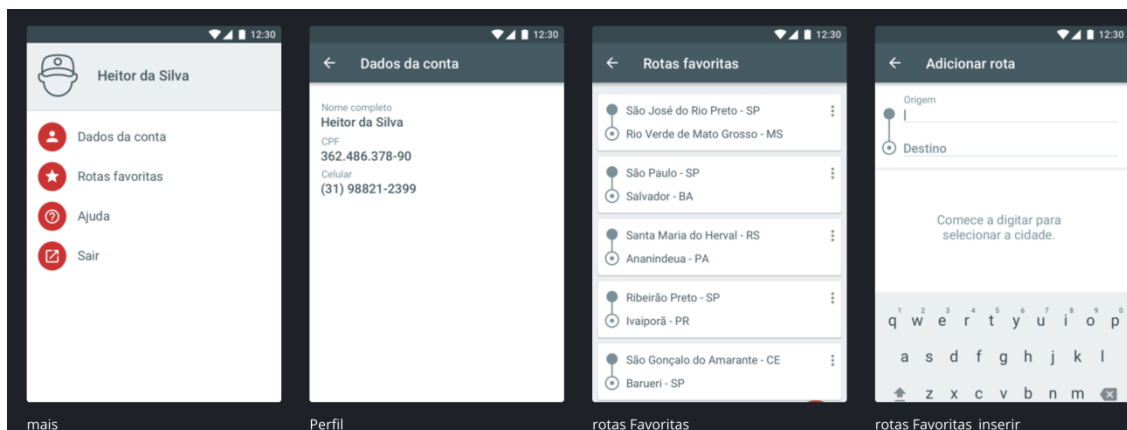


Ilustración 23 – Pantallas de perfil del usuario

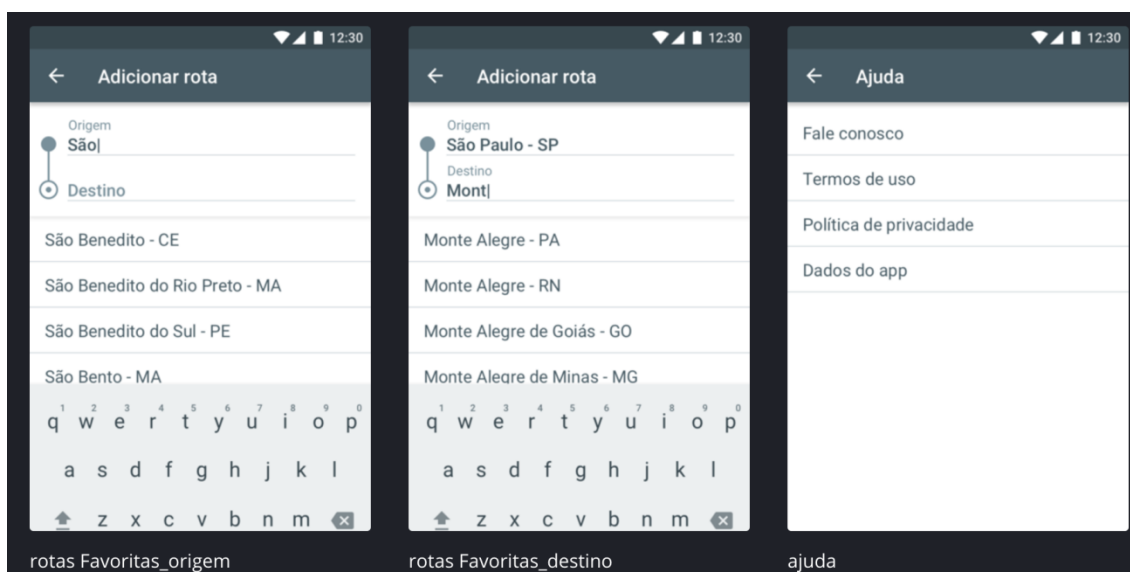


Ilustración 24 – Pantallas de perfil del usuario (cont.)

Componentes (11 pantallas):

En las siguientes pantallas se observan diferentes componentes que posee la app (véase Ilustraciones 25, 26 y 27):

- La primera pantalla muestra la disposición de las cargas en forma de cuadros.
- La segunda pantalla muestra lo que se visualiza cuando no hay rutas disponibles.
- La tercera pantalla muestra lo que se visualiza cuando no hay pedidos de transporte de cargas.
- La cuarta pantalla muestra lo que se visualiza cuando no hay historial de transportes realizados.
- La quinta pantalla muestra lo que se visualiza cuando no hay rutas favoritas agregadas.
- La sexta pantalla muestra cómo llamar a un contacto dentro de la aplicación.
- La séptima pantalla muestra cartel de confirmación de inicio del transporte.
- La octava y novena pantalla muestran cartel de confirmación para que la aplicación pueda enviar o recibir SMS y permitir acceder a la localización del dispositivo móvil.
- La décima pantalla muestra datos de la aplicación.
- La onceava pantalla sólo muestra el ícono de eliminar.

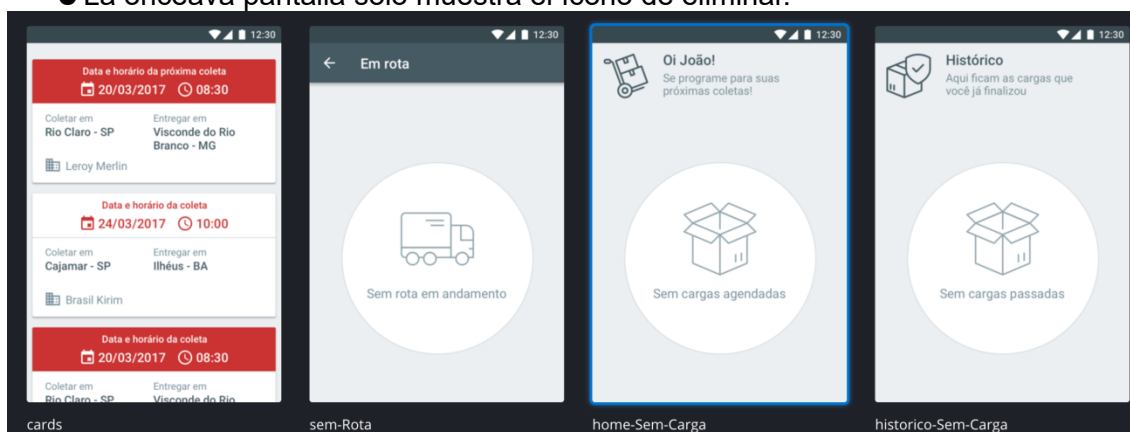
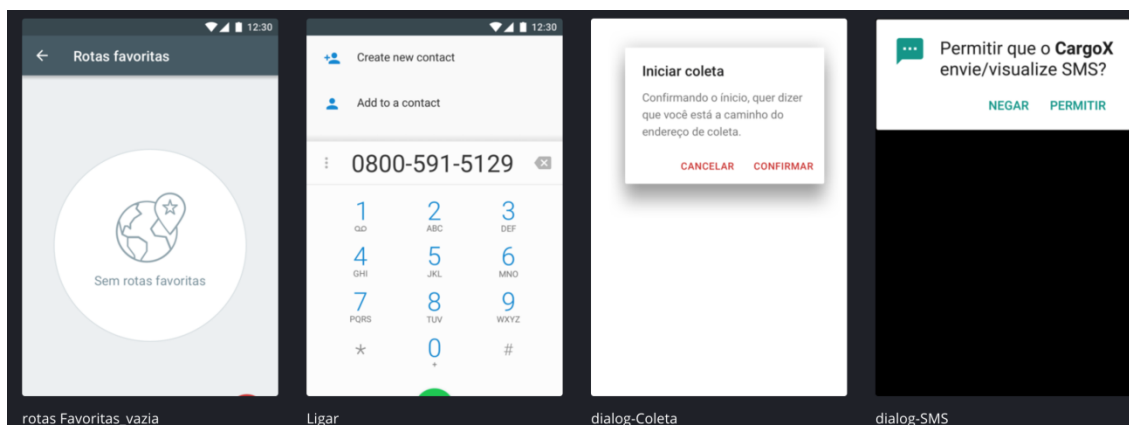
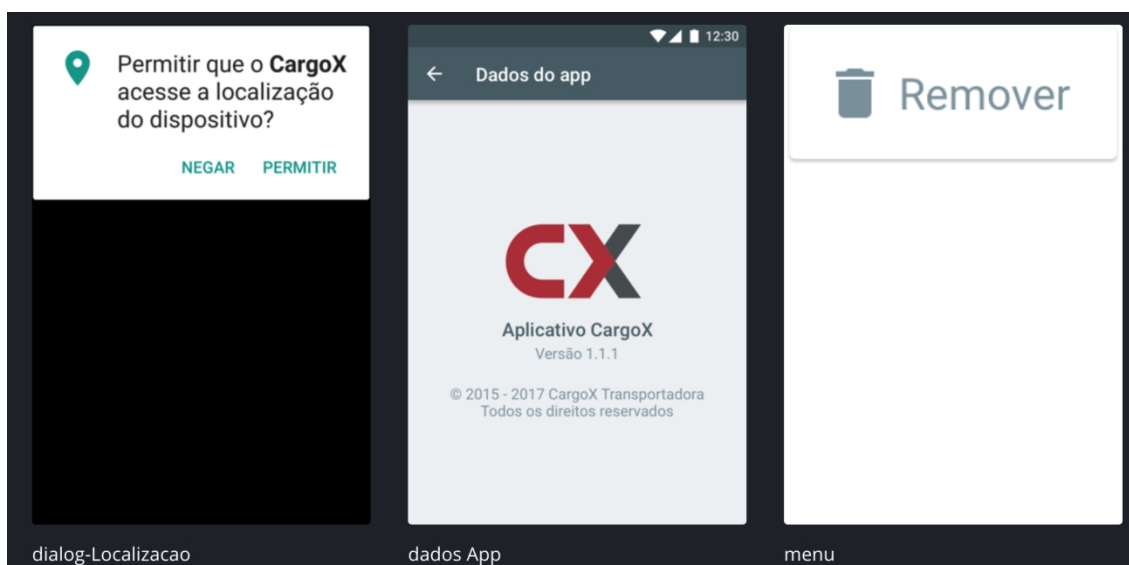


Ilustración 25 – Pantallas de componentes



**Ilustración 26 – Pantallas de componentes (cont. 1)**



**Ilustración 27 – Pantallas de componentes (cont. 2)**

#### 4.3.3.- Interacción del producto con otros sistemas

El producto de software utiliza librerías externas para, por ejemplo, determinar recorridos utilizando el GPS (librería de Google), fijar precios, consultar rutas, peajes (librería KMM, brasilera), entre otras. Todas las librerías son enlazadas a la parte de backend, la cual posee una API para todo el sistema en sí y otra separada que se encarga del seguimiento o trackeo del transportista al que le envían las coordenadas.

#### 4.3.4.- Seguridad del producto de software

La seguridad del producto es controlada por parte de los mismos desarrolladores, protegiendo el acceso al sistema por autenticación. Cuando los usuarios desean ingresar al sistema deben autenticarse por medio de un número de teléfono y un número CPF.

De momento la aplicación sólo posee un rol, el de transportista, y es por eso que no es necesario controlar los permisos.

#### 4.4.- Organización desarrolladora del producto de software

##### 4.4.1.- Descripción de la organización

La organización propietaria y desarrolladora de la aplicación móvil CargoX, es Sontra. Esta empresa, creada en 2013 en Brasil, lanzó la aplicación SontraCargo, la cual se encargaba de conectar fletes o cargas con transportistas autónomos. Brasil posee un mercado grandes en lo que concierne a transporte de camiones, y es por esto que fue una aplicación muy utilizada en ese país. Actualmente, Sontra ha desarrollado la aplicación móvil, denominada CargoX (la utilizada en este trabajo) que posee los mismos objetivos que la anterior, pero es más intuitiva, posee una interfaz más amigable, fácil de utilizar y agrega nuevas funcionalidades, dejando obsoleta a la antigua SontraCargo [L].

Esta organización posee una sub-empresa en Argentina, más específicamente en la ciudad de Buenos Aires, responsable del desarrollo de CargoX. Esta empresa en Argentina se llama como la aplicación móvil, CargoX, y se creó en 2016.

La empresa en total cuenta, hoy en día, con unas 170 personas. Los que están involucrados en el producto son en total 22 personas, entre ellos, tres desarrolladores Android, diez PHP, cinco de QA<sup>5</sup>, tres socios fundadores, un CTO<sup>6</sup> entre otros.

##### 4.4.2.- Infraestructura

Para desarrollar la app cada desarrollador utiliza su equipamiento. Todos deben utilizar PHP con MySQL y almacenen en servidores amazon preferentemente con sistema operativo Linux.

#### 4.5.- Conclusiones

Durante toda esta sección se describió en detalle el producto de software a evaluar (objetivos, alcance, su seguridad, interacción con otros sistemas, etc), desarrollada para dispositivos móviles con sistema operativo Android.

Se mostraron mediante ilustraciones las pantallas que posee dicha aplicación, las cuales van a ser de gran utilidad para recabar datos a la hora de realizar los cálculos de las mediciones para las métricas seleccionadas en la evaluación de calidad que se hará en el *CAPITULO 8.- EJECUCION DE LA EVALUACION*.

Se mencionaron también algunos aspectos de la organización que desarrolló CargoX, llamada Sontra. No se ahondó demasiado en este punto ya que no es vital conocer en detalle la empresa para este trabajo.

---

<sup>5</sup>La sigla QA significa Quality Assurance, o aseguramiento de la calidad. Se trata de un conjunto de actividades de evaluación de las distintas etapas del proceso de desarrollo para garantizar que el producto final sea de calidad.

<sup>6</sup>El director de tecnología (del inglés chief technical officer o chief technology officer, abreviado como CTO) es una posición ejecutiva dentro de una organización en el que la persona que ostenta el título se concentra en asuntos tecnológicos y científicos.

---

## CAPITULO 5.- REQUISITOS DE LA EVALUACION

---

### 5.1.- Propósito de la evaluación

El propósito de la evaluación de calidad es obtener el nivel de calidad de la aplicación móvil seleccionada para poder identificar sus debilidades y fortalezas de una manera formalmente mensurable, en el marco de un proceso de mejora continua.

Teniendo en cuenta el objetivo de la evaluación, los resultados y las conclusiones deberán indicar aspectos a mejorar en el producto de software, los cuales deberán disparar acciones correctivas planificadas y ejecutadas durante un período de tiempo especificado, al fin del cual se realizaría una nueva evaluación, de modo de ir cubriendo otros aspectos del producto. Este proceso de mejora continua se repetiría aumentando la calidad a través del tiempo.

### 5.2.- Motivación de la evaluación

Lo que motiva la realización de esta evaluación de calidad sobre la aplicación móvil CargoX es:

- Necesidad de conocer las debilidades y fortalezas que posee la aplicación, en el marco de un proceso de mejora continua.
- Necesidad de conocer en detalle las debilidades de CargoX para poder identificar las acciones de mejora a realizar en el marco de un proceso de mejora continua.
- Necesidad de estudiar características de calidad como *usabilidad* y *seguridad* teniendo en cuenta la importancia y la consideración que se les da hoy en día en el contexto de una aplicación móvil, tomando como base las opiniones de diferentes profesionales [M, N, O].

### 5.3.- Tipo de producto a evaluar

El producto de software seleccionado es una aplicación móvil desarrollada para Android versión 4.0 en adelante, llamada CargoX. Es un producto terminado, más allá de que actualmente se le están realizando modificaciones e implementando nuevas funcionalidades. El mismo ya fue lanzado al mercado, siendo utilizado en este momento por aproximadamente 1300 usuarios.

Esta evaluación se realizará sobre la versión 1.2.1 de la app.

### 5.4.- Modelo de calidad

El modelo de calidad que se va a utilizar en la evaluación de calidad para el producto de software seleccionado se encuentra definido en la norma ISO/IEC 25010-*System and software quality models*, mientras que el proceso de evaluación se encuentra en la norma ISO/IEC 25040-*Evaluation process*. Teniendo en cuenta el tipo de producto de software a evaluar (producto final), se tendrá en cuenta la calidad externa para la evaluación de calidad, y se aplicarán parte de las métricas externas que define la norma ISO/IEC 25023-*Measurement of system and software product quality*.

La norma ISO/IEC 25010-*System and software quality models* especifica ocho características de calidad, las cuales son subdivididas en subcaracterísticas, que fueron claramente enumeradas y definidas en la sección 2.3.4.1.- *Norma ISO/IEC 25010-System and software quality models*.

Cabe aclarar que se descarta la *ISO/IEC 25042 – Process for developers*, porque la evaluación, si bien se solicitan ciertos datos o reportes al equipo de desarrollo de la aplicación móvil CargoX, no se lleva a cabo en paralelo al desarrollo del producto, sino que se lleva a cabo sobre un producto final. También es descartada la *ISO/IEC 25044 – Process for evaluators*, ya que la evaluación no la realizan evaluadores independientes, sino que la realiza un evaluador que se enfocará en el punto de vista de aquellas personas que adquieren el producto, o compran el producto y lo utilizan (usuarios o clientes). Es por esto último que se va a utilizar la *ISO/IEC 25043 – Process for acquirers* para realizar la evaluación de calidad sobre la aplicación móvil CargoX.

#### 5.4.1.- Selección de las características a evaluar

En el caso concreto de la aplicación móvil seleccionada, dentro de las características que más importan, tanto a los usuarios de la misma como a los desarrolladores y propietarios, se encuentran las de *usabilidad* y *seguridad*, y por este motivo es que se eligen estas dos para ejecutar la evaluación de calidad, con sus respectivas subcaracterísticas descritas a continuación.

##### 5.4.1.1.- Importancia de la evaluación de la característica *usabilidad* para aplicaciones móviles

La *usabilidad* es considerada uno de los factores más importantes dentro de la calidad de un producto de software, más específicamente de una aplicación móvil. Con la aparición, el uso masivo y el crecimiento de los dispositivos móviles, especialmente los smartphones, la medición de *usabilidad* en aplicaciones móviles se tornó un tema de investigación. Particularmente la *usabilidad* de una “app” se refiere a la facilidad con que los usuarios pueden utilizar la misma para alcanzar un objetivo concreto.

##### 5.4.1.1.1- Importancia de la evaluación de la subcaracterística *inteligibilidad* para aplicaciones móviles

Esta subcaracterística hace referencia a la capacidad por parte de la app de hacer entender su adecuación para las necesidades del usuario.

##### 5.4.1.1.2.- Importancia de la evaluación de la subcaracterística *aprendizaje* para aplicaciones móviles

La subcaracterística *aprendizaje* hace referencia a la forma en que el producto de software permite al usuario aprender su uso. Es de suma importancia evaluar esta subcaracterística, ya que en el peor de los casos, puede ocurrir que no exista la facilidad de aprendizaje, haciendo que la aplicación pierda usuarios por no poder realizar las tareas que brinda la app.



#### 5.4.1.1.3.- Importancia de la evaluación de la subcaracterística *operabilidad* para aplicaciones móviles

La subcaracterística *operabilidad* hace referencia a la manera en que el producto de software permite al usuario operarlo y controlarlo. Dicho esto se observa claramente porque es importante que esta subcaracterística sea evaluada: el hecho de que la aplicación móvil no pueda ser controlada u operada con facilidad presupone un alto riesgo para la misma, haciendo que cada vez sean menos los usuarios que la descargan.

#### 5.4.1.1.4.- Importancia de la evaluación de la subcaracterística *protección frente a errores de usuario* para aplicaciones móviles

Esta subcaracterística hace referencia a la capacidad del sistema para proteger a los usuarios de cometer errores. Si se producen, es importante que se den a conocer al usuario de forma rápida y clara, además de ofrecer algún mecanismo para recuperarse de ese error.

#### 5.4.1.1.5.- Importancia de la evaluación de la subcaracterística *estética* para aplicaciones móviles

La subcaracterística *estética* hace referencia a la capacidad de la interfaz de usuario de agrandar y satisfacer la interacción con el usuario. Esta subcaracterística es muy importante para ser evaluada, ya que los usuarios se fijan mucho en la manera en que están predispuestas las funcionalidades dentro de la aplicación, o la forma en que se encuentran los iconos, colores, etc, haciendo más fácil su navegación por la app y satisfaciendo sus necesidades.

#### 5.4.1.1.6.- Importancia de la evaluación de la subcaracterística *accesibilidad* para aplicaciones móviles

La subcaracterística *accesibilidad* hace referencia a la capacidad del producto para ser utilizado por usuarios con determinadas características y discapacidades, permitiendo así que todas las personas puedan utilizar la aplicación móvil sin problemas, incluso aquellas con limitaciones físicas, visuales, auditivas o de otra índole, y sumando más usuarios.

#### 5.4.1.2.- Importancia de la evaluación de la característica *seguridad* para aplicaciones móviles

El comportamiento de una aplicación móvil preocupa al usuario desde el punto de vista de la *seguridad*, ya que estarán disponibles en dispositivos que pueden ser susceptibles de robo o de uso fraudulento. En este sentido, resulta imprescindible asegurar, entre otros factores, que la app mantenga la confidencialidad de los datos privados, la verificación del usuario, los diferentes perfiles que puede tener un usuario dentro de la misma aplicación o el autoguardado de un histórico de operaciones o actividades realizadas.

#### 5.4.1.2.1.- Importancia de la evaluación de la subcaracterística *integridad* para aplicaciones móviles

Es importante evaluar esta subcaracterística para determinar el grado en que un sistema, producto o componente impide el acceso no autorizado o la modificación de programas o datos de computadora.

#### 5.4.1.2.2.- Importancia de la evaluación de la subcaracterística *autenticidad* para aplicaciones móviles

Las medidas de autenticidad se utilizan para evaluar el grado en que se puede demostrar que la identidad de un sujeto o recurso es la que se atribuye, siendo muy importante evaluar esta subcaracterística con la finalidad de conocer si la aplicación móvil posee reglas o medidas de autenticidad adecuadas como para darse cuenta si un usuario logeado en el sistema es quien dice ser.

### 5.5.- Requerimientos de calidad

En esta sección se mostrarán los requerimientos de calidad, es decir, se describirán los requisitos que van a ser evaluados sobre la aplicación móvil con las características y subcaracterísticas de calidad relacionadas con cada uno, con la finalidad de ayudar a la empresa Sontra a describir con precisión lo que desea obtener. Cuando se haga referencia a dichas características o subcaracterísticas durante este trabajo a partir de ahora, se hará referencia a los requerimientos de calidad a evaluar y viceversa.

ID Requerimiento	Característica	Subcaracterística	Descripción
REQUERIMIENTO 1	<i>Usabilidad</i>	Estética	Evaluar la capacidad de la interfaz de usuario de la aplicación para la interacción y navegación del usuario en la misma.
REQUERIMIENTO 2	<i>Usabilidad</i>	Inteligibilidad	Evaluar la capacidad de la aplicación móvil para reconocer su adecuación.
REQUERIMIENTO 3	<i>Usabilidad</i>	Operabilidad	Evaluar la capacidad de la aplicación móvil de ser controlada con facilidad.

REQUERIMIENTO 4	<i>Usabilidad</i>	Aprendizaje	Evaluar el nivel de facilidad de aprendizaje de la aplicación móvil.
REQUERIMIENTO 5	<i>Usabilidad</i>	Protección frente a errores de usuario	Evaluar el nivel de protección y recuperación contra errores que pueden cometer los usuarios que provee la aplicación móvil.
REQUERIMIENTO 6	<i>Usabilidad</i>	Accesibilidad	Evaluar el nivel de accesibilidad que presenta la aplicación móvil.
REQUERIMIENTO 7	<i>Usabilidad</i>	Estética	Evaluar cantidad de iconos, estilos, etc, dentro de la aplicación móvil.
REQUERIMIENTO 8	<i>Seguridad</i>	Integridad	Evaluar impedimentos sobre accesos no autorizados.
REQUERIMIENTO 9	<i>Seguridad</i>	Integridad	Evaluar impedimentos sobre modificaciones de datos o programas de computadora.
REQUERIMIENTO 10	<i>Seguridad</i>	Autenticidad	Evaluar la capacidad de demostración de la identidad de un sujeto o recurso.

**Tabla 8 - Requerimientos de calidad a evaluar**

### 5.6.- Grado de rigurosidad

El grado de rigurosidad con el que se realiza la evaluación de calidad es alto, ya que se trata de una aplicación móvil que hace poco tiempo fue lanzada en el mercado (un año) y aún son pocos los usuarios que la utilizan (aproximadamente 1300). Se desea encontrar varios aspectos a mejorar, teniendo en cuenta que es la primera evaluación de calidad que se le realiza al producto desde que salió al mercado.

En la tabla 9 se configura cada requerimiento a evaluar, mencionados con anterioridad, con el nivel de evaluación correspondiente, de acuerdo a la clasificación de los niveles de evaluación dados por la norma ISO/IEC 25044-*Process for evaluators*, donde se define una jerarquía de niveles desde la A (más alta), hasta la D (más baja). El nivel A implica usar técnicas de evaluación más exigentes, y la exigencia va bajando gradualmente hasta el nivel D.

ID del requerimiento	Aspectos de seguridad (física)	Aspectos económicos	Aspectos de seguridad (acceso)	Aspectos relacionados con el medio ambiente
REQUERIMIENTO 1	N/A	D	N/A	N/A
REQUERIMIENTO 2	N/A	B	N/A	N/A
REQUERIMIENTO 3	N/A	A	N/A	N/A
REQUERIMIENTO 4	N/A	B	N/A	N/A
REQUERIMIENTO 5	N/A	A	N/A	N/A
REQUERIMIENTO 6	N/A	C	N/A	N/A
REQUERIMIENTO 7	N/A	D	N/A	N/A
REQUERIMIENTO 8	N/A	B	A	N/A
REQUERIMIENTO 9	N/A	A	B	N/A
REQUERIMIENTO 10	N/A	B	B	N/A

**Tabla 9 - Nivel de evaluación para cada requerimiento de calidad**

### 5.7.- Grado de confidencialidad

La confidencialidad de todos los componentes del producto de software y documentos relacionados con éste, es decir toda la información y elementos de entrada brindados por la organización Sontra para la evaluación del producto CargoX, y de la documentación de la evaluación incluyendo los registros y el reporte de la misma debe estar protegida por el evaluador.

A su vez, Sontra protege la confidencialidad de todos los reportes, información, documentos, etc, que le parece apropiado, pudiendo brindar al evaluador solo lo que el mismo requiriese y lo que la empresa desee compartir (no la totalidad).

## 5.8.- Conclusiones

Las características más importantes a la hora de utilizar una app (según opiniones de diversos profesionales), son las de *usabilidad*, que es la medida en la cual un producto puede ser usado por usuarios determinados para conseguir objetivos específicos con efectividad, eficiencia y satisfacción en un contexto de uso establecido, y *seguridad*, que es la medida de protección de la integridad y privacidad de la información (o datos) almacenada en un sistema informático [M, N, O]. La empresa propietaria de CargoX y sus desarrolladores coincidieron en la selección de estas dos características para ser evaluadas, teniendo en cuenta también cada una de sus subcaracterísticas.

A partir de los requerimientos de calidad se pudo observar qué es lo que el cliente desea exactamente o cuál es el objetivo del cliente para llevar a cabo la evaluación de calidad.

Como conclusión, la evaluación se realiza con un alto grado de rigurosidad, ya que CargoX es nueva y necesita mejorar ciertos aspectos para poder competir con otras aplicaciones de su tipo y lograr posicionarse en el mercado.

## CAPITULO 6.- ESPECIFICACION DE LA EVALUACION

---

### 6.1.- Alcance de la evaluación

La evaluación de calidad abarca el producto de software CargoX, aplicación móvil desarrollada para Android 4.0 en adelante, incluyendo la totalidad del sistema, exceptuando componentes que no son parte del mismo, como por ejemplo las API's a las cuales se conecta o el motor de base de datos que utiliza (MySQL), o toda la parte del sistema que se ejecuta mediante la utilización del dashboard.

### 6.2.- Selección de métricas

Las características a evaluar sobre el producto de software seleccionado (CargoX) se dimensionan utilizando métricas. En la sección 5.4.- *Modelo de calidad* se determinó el modelo de calidad a utilizar, y se detallaron todas aquellas características más importantes (*usabilidad y seguridad*) y subcaracterísticas que van a ser evaluadas más adelante. También se indicó que se tomará en cuenta la calidad externa para la evaluación de calidad, y se aplicarán parte de las métricas externas que define la norma ISO/IEC 25023-*Measurement of system and software product quality*. En base a todo esto, se realiza la selección de métricas a aplicar sobre la aplicación móvil.

Como la aplicación a la cuál se le va a realizar la evaluación de calidad es prácticamente nueva, se tomará en cuenta para el comportamiento del producto desde el momento en que fue lanzado al mercado hasta el día de en que se realiza esta evaluación (de Noviembre del 2016 hasta Febrero del 2018). Este rango de tiempo servirá como entrada para determinadas características.

En las siguientes secciones se determinarán las métricas para evaluar los requerimientos de calidad definidos en 5.5.- *Requerimientos de calidad*.

#### 6.2.1.- Métricas seleccionadas para evaluar la *usabilidad*

Subcaracterística	Requerimiento a evaluar	Métrica seleccionada	Motivo de aplicación
Estética	REQUERIMIENTO 1  REQUERIMIENTO 7	- <i>Personalización de los elementos de la interfaz de usuario.</i>	Con esta métricas seleccionada se pretende evaluar cuán detallada es la interfaz de usuario para su utilización y el nivel de contenido de colores, estilos, iconos, etc dentro de la aplicación.

Inteligibilidad	REQUERIMIENTO 2	<i>-Capacidad de demostración.</i>	Con estas métricas se pretende evaluar si el nivel de adecuación es el correcto para el usuario, es decir, si el producto de software tiene la capacidad de hacer entender al usuario si fue creado para sus necesidades.
Operabilidad	REQUERIMIENTO 3	<i>-Claridad del mensaje. -Posibilidad de personalización.</i>	Con estas métricas seleccionadas se pretende evaluar la capacidad que posee la aplicación móvil de ser operada con facilidad por parte del usuario.
Aprendizaje	REQUERIMIENTO 4	<i>-Campos de entrada con valores predeterminados.</i>	Con estas métricas se desea evaluar cuán fácil puede el usuario utilizar la app, es decir, cual es la capacidad que tiene la aplicación que permite al usuario aprender sobre el uso de la misma.



Protección frente a errores de usuario	REQUERIMIENTO 5	<i>-Comprobación de errores de entrada de usuario.</i>	Con esta métrica se pretende evaluar si el sistema de protección frente a errores de usuario es el adecuado o no, es decir, si la capacidad del sistema para proteger a los usuarios de cometer errores es correcto o no lo es.
Accesibilidad	REQUERIMIENTO 6	<i>-Número de idiomas soportados.</i>	Con estas métricas se pretende evaluar si la aplicación posee la capacidad de ser accesible, es decir, permitir que usuarios con discapacidades o capacidades reducidas puedan utilizar la app sin inconvenientes.

**Tabla 10 - Métricas seleccionadas para evaluar la *usabilidad***

### 6.2.2.- Métricas seleccionadas para evaluar la *seguridad*

La tabla 11 muestra las métricas seleccionadas para evaluar los requerimientos de calidad relacionados a la *seguridad* del producto:

Subcaracterística	Requerimiento a evaluar	Métrica seleccionada	Motivo
Integridad	REQUERIMIENTO 8 REQUERIMIENTO 9	<i>-Prevención de corrupción de datos.</i>	Con esta métrica se pretende evaluar en qué medida se previene la corrupción de datos teniendo en cuenta los accesos de un mismo usuario dentro del sistema CargoX.
Autenticidad	REQUERIMIENTO 10	<i>-Protocolos de autenticación.</i>	Con esta métrica se pretende evaluar qué tan bien autentica el sistema la identidad de un sujeto o recurso.

Tabla 11 - Métricas seleccionadas para evaluar la *seguridad* del producto

### 6.2.3.- Descripción de las métricas seleccionadas

En la tabla 12 se describen cada una de las métricas seleccionadas con la finalidad de comprender mejor cada una de ellas, indicando, también, su fórmula para realizar las mediciones.

Métrica	Descripción	Fórmula	Observaciones
<i>Personalización de los elementos de la interfaz de usuario</i>	¿Qué proporción de elementos de interfaz de usuario se pueden personalizar en apariencia?	$X = A / B$ A = Número de elementos de la interfaz de usuario que se pueden personalizar. B = Número total de elementos de la interfaz de usuario	Considerar el uso del producto de software entre Noviembre del 2016 y Febrero del 2018, utilizando un entero para el número de elementos de la interfaz del usuario y para la totalidad de los mismos.

<i>Capacidad de demostración</i>	¿Qué proporción de funciones tiene capacidad de demostración para que los usuarios reconozcan la idoneidad?	$X = A / B$ A = Número de funciones implementadas con capacidad de demostración B = Número total de funciones en la descripción del producto	Considerar el uso del producto de software entre Noviembre del 2016 y Febrero del 2018, utilizando un entero para el número de funciones con capacidad de demostración y para su totalidad.
<i>Claridad del mensaje</i>	¿Qué parte de los mensajes de un sistema se puede entender fácilmente?	$X = A / B$ A = Número de mensajes que son entendidos por los usuarios simulados B = Número total de mensajes implementados	Considerar el uso del producto de software entre Noviembre del 2016 y Febrero del 2018, utilizando un entero para los número de mensajes entendibles y para la totalidad de mensajes implementados.
<i>Posibilidad de personalización</i>	¿Qué porción de funciones y procedimientos operativos puede personalizar un usuario para su conveniencia?	$X = A / B$ A = Número de funciones y procedimientos operativos que pueden personalizarse para la comodidad del usuario B = Número total de funciones y procedimientos operativos implementados	Considerar el uso del producto de software entre Noviembre del 2016 y Febrero del 2018, utilizando un entero para el número de funciones y procedimientos operativos personalizables y para su totalidad.

<i>Campos de entrada con valores predeterminados</i>	¿Qué proporción de campos de entrada que podrían tener valores por defecto útiles se rellenan automáticamente con valores predeterminados?	$X = A / B$ A = Número de campos de entrada que podrían tener valores por defecto útiles B = Número de campos de entrada total.	Considerar el uso del producto de software entre Noviembre del 2016 y Febrero del 2018, utilizando un entero para el número de campos de entrada con valores por defecto útiles y para su totalidad.
<i>Comprobación de errores de entrada de usuario</i>	¿Qué proporción de errores detectables de entrada del usuario es identificada por el sistema y claramente descrita para facilitar la corrección del error?	$X = A / B$ A = Número de errores identificados y claramente descritos para corregir los errores de entrada del usuario por el sistema B = Número total de errores detectables de entrada de usuario	Considerar el uso del producto de software durante la primera quincena del mes de Febrero del 2018, utilizando un entero para el número de errores y para su totalidad.
<i>Número de idiomas soportados</i>	¿Cuántos idiomas diferentes se admiten?	$X = A / B$ A = Número de idiomas realmente soportados B = Número necesario de idiomas a ser soportados definidos en la descripción del producto	Considerar el uso del producto de software entre Noviembre del 2016 y Febrero del 2018, utilizando un entero para el número de idiomas soportados y los realmente soportados.

<i>Prevención de corrupción de datos</i>	¿En qué medida se previene la corrupción de datos?	$X = A / B$ A = Número de accesos que realmente corrompen los datos. B = Número de accesos donde se debe prevenir daño o rotura de datos.	Considerar los primeros 15 días del mes de Febrero del 2018, teniendo en cuenta al número de accesos (de un mismo usuario) que corrompen los datos como un entero. El valor del resultado varía de 0 a infinito. Por lo general, cuanto más cerca de 0 es mejor.
<i>Protocolos de autenticación</i>	¿Qué tan bien autentica el sistema la identidad de un sujeto o recurso?	$X = A / B$ A = Número de protocolos de autenticación proporcionados (p. Ej., ID de usuario / contraseña o tarjeta IC, etc). B = Número de protocolos de autenticación requeridos en la especificación.	Considerar el uso del producto de software entre Noviembre del 2016 y Febrero del 2018, utilizando un número entero para el número de protocolos de autenticación proporcionados y visibles para el usuario de la app.

Tabla 12 - Descripción de las métricas seleccionadas

### 6.3.- Niveles de rating

Para cada métrica se definirá un rango de valores, que, según la importancia para CargoX y sus desarrolladores, variará de un porcentaje mínimo de aceptación a uno no máximo (véase observaciones en la tabla 13). Esto determinará la categoría o nivel de rating del resultado de acuerdo al grado de satisfacción de los requerimientos de calidad (definidos en 5.5.-*Requerimientos de calidad*). Puede decirse que se definirá un criterio de evaluación a nivel de las métricas, para luego desarrollar en 6.4 el criterio que determinará el resultado de la evaluación.

Los rangos de valores definidos son los siguientes:

- Rango de no aceptación: es el rango de valores que se consideran no aceptables para la métrica. Un resultado en este rango se categoriza como no aceptable.

- Rango de aceptación: rango de valores en el cual debe caer el resultado de la métrica para ser categorizado como aceptable. Puede estar conformado por un solo valor, en ese caso se puede denominar valor de aceptación.
- Rango planeado: rango de valores entre los cuales se espera que se encuentre el resultado de la métrica. Este rango debe considerar valores iguales o mejores a los que componen el rango de aceptación y puede estar conformado por un solo valor, en ese caso se lo puede denominar valor planeado.
- Rango de exceso: es el rango de valores que se consideran superiores a los valores del rango planeado de la métrica. Un resultado en este rango se categoriza como excede.

Un ejemplo de aplicación de ésta categorización puede ser el siguiente: sea  $M$  una métrica donde,

1.  $[V_0, V_n]$  es su rango de valores,
2. Su interpretación indica que el mejor valor es el más cercano a  $V_n$ .

Se seleccionan los valores  $V_i, V_j$  y  $V_k$  pertenecientes al rango de valores de la métrica, donde se tiene que  $V_x < V_i < V_j < V_k < V_n$  y se define la siguiente categorización para el nivel de rating:

- Rango de no aceptación:  $[V_0, V_i)$ , formado por resultados  $X$  de la métrica tales que  $V_0 \leq X < V_i$
- Rango de aceptación:  $[V_i, V_j]$ , formado por resultados  $X$  de la métrica tales que  $V_i \leq X \leq V_j$
- Rango planeado:  $(V_j, V_k]$ , formado por resultados  $X$  de la métrica tales que  $V_j < X \leq V_k$
- Rango de exceso:  $(V_k, V_n]$ , formado por resultados  $X$  de la métrica tales que  $V_k < X \leq V_n$

La ilustración 28 muestra un gráfico de la categorización definida:

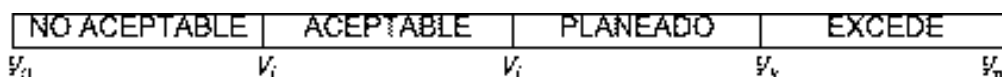


Ilustración 28 – Categorización según rango de valores

Entonces, siendo  $X$  el resultado de la métrica, se aplica la categorización:

- si  $X < V_i$  entonces el nivel de rating del resultado de la métrica es no aceptable.
- si  $X \in [V_i, V_j]$ , entonces el nivel de rating del resultado de la métrica es aceptable.
- si  $X \in (V_j, V_k]$ , entonces el nivel de rating del resultado de la métrica es planeado.
- si  $X > V_k$ , entonces el nivel de rating del resultado de la métrica es excede.

Este mismo criterio puede adaptarse para el caso en el cual el resultado sea mejor si está más cercano a  $V_0$ . También, de acuerdo al caso dependerá cuales extremos de los rangos serán abiertos o cerrados. Vale aclarar también, que indistintamente se puede hacer referencia al nivel de rating del resultado de la métrica o al nivel de rating de la métrica significando lo mismo en ambos casos.

Finalmente, cada característica y subcaracterística también serán categorizadas con un nivel de rating de acuerdo a lo explicado en 6.4 *Criterios de evaluación*.

### 6.3.1.- Niveles de rating para evaluar los requerimientos de calidad a través de las métricas

En la siguiente tabla (tabla 13) se muestra cómo determinar el nivel de rating para cada métrica, en base a su resultado y así establecer el grado de satisfacción con el requerimiento de calidad correspondiente. Se tienen en cuenta los cuatro tipos de rangos mencionados anteriormente en la sección 6.3.-*Niveles de rating*. Cada una de las correspondencias de las métricas con su nivel de rating es reconocido como objetivo de calidad del producto, indicando así, el grado de satisfacción de los requerimientos de calidad.

<b>Métrica</b>	<b>Resultados esperados sobre categorización de nivel de rating</b>	<b>Observaciones</b>
<i>Personalización de los elementos de la interfaz de usuario</i>	Rango no aceptable: menor a 50%. Rango aceptable: entre 50% y 75%. Rango planeado: mayor a 75%, hasta 95%. Rango excede: mayor a 95%.	Se espera que el resultado de la métrica sea mayor al 75%, es decir, ubicandose en el rango planeado, aunque si el resultado entra dentro del rango excede sería óptimo. Para obtener este porcentaje se tiene en cuenta la función $X = A / B$ definida en la tabla 12 para esta métrica.
<i>Capacidad de demostración</i>	Rango no aceptable: menor a 50%. Rango aceptable: entre 50% y 75%. Rango planeado: mayor a 75%, hasta 95%. Rango excede: mayor a 95%.	En este caso, se espera que el número de funciones implementadas con capacidad de demostración sea del 75% o mayor (entrando en el rango planeado) sobre el total de funciones, siendo mejor si este porcentaje supera el 95%, alcanzando el rango de excede. No es aceptable si este porcentaje es menor al 50%, y es aceptable si está entre 50% y 75%. La función que se utiliza para obtener los porcentajes es $X = A / B$ , siendo A el número de funciones implementadas con capacidad de demostración y B el número total de funciones.



<i>Claridad del mensaje</i>	<p>Rango no aceptable: menor a 50%.</p> <p>Rango aceptable: entre 50% y 75%.</p> <p>Rango planeado: mayor a 75%, hasta 95%.</p> <p>Rango excede: mayor a 95%.</p>	<p>Se espera un rango planeado (entre 75% y 95%) o mayor a 95% (rango excede) para evitar mensajes que no sean claros para el usuario dentro de la aplicación móvil, entrando en este caso en un rango no aceptable (menor a 50% de mensajes claros sobre la totalidad de mensajes dentro de la aplicación). Un rango aceptable sería si el porcentaje de estos mensajes cae dentro del rango entre 50% y 75%.</p> <p>Para obtener este porcentaje se tiene en cuenta la función <math>X = A / B</math> definida en la tabla 12 para esta métrica.</p>
<i>Posibilidad de personalización</i>	<p>Rango no aceptable: menor a 50%.</p> <p>Rango aceptable: entre 50% y 75%.</p> <p>Rango planeado: mayor a 75%, hasta 95%.</p> <p>Rango excede: mayor a 95%.</p>	<p>Se espera que el porcentaje de posibilidad de personalización esté entre 75% y 95%, y mejor aún si este porcentaje supera el 95%. Es importante que el usuario pueda personalizar funciones y procedimientos dentro de la app, y es por eso que se espera un porcentaje alto, tratando siempre de evitar el rango de no aceptación.</p> <p>Para obtener este porcentaje se tiene en cuenta la función <math>X = A / B</math> definida en la tabla 12 para esta métrica.</p>
<i>Campos de entrada con valores predeterminados</i>	<p>Rango no aceptable: menor a 50%.</p> <p>Rango aceptable: entre 50% y 75%.</p> <p>Rango planeado: mayor a 75%, hasta 95%.</p> <p>Rango excede: mayor a 95%</p>	<p>Es importante para el usuario observar en los campos de entrada los valores predeterminados y así poder completarlos con mayor facilidad. Por esto se espera que el resultado supere el 75%, obtenido a partir de la función <math>X = A / B</math> definida en la tabla 12.</p>

<i>Comprobación de errores de entrada de usuario</i>	<p>Rango no aceptable: menor a 65%.</p> <p>Rango aceptable: entre 65% y 85%.</p> <p>Rango planeado: mayor a 85%, hasta 98%.</p> <p>Rango excede: mayor a 98%.</p>	<p>No pueden permitirse errores de esta índole. Es por esto que se espera un alto porcentaje de correctitud, más específicamente, superior a 85%.</p> <p>Para obtener este porcentaje se tiene en cuenta la función <math>X = A / B</math> definida en la tabla 12 para esta métrica.</p>
<i>Número de idiomas soportados</i>	<p>Rango no aceptable: menor a 50%.</p> <p>Rango aceptable: entre 50% y 75%.</p> <p>Rango planeado: mayor a 75%, hasta 95%.</p> <p>Rango excede: mayor a 95%.</p>	<p>Un rango no aceptable es si el resultado de aplicar la métrica es menor al 50%. Es aceptable si está entre 50% y 75%. Entra en rango planeado si es mayor a 75% y menor o igual a 95% y en excede si es mayor a 95%. Estos dos últimos rangos son los esperados para esta métrica.</p>
<i>Prevención de corrupción de datos</i>	<p>Rango no aceptable: mayor a 35%.</p> <p>Rango aceptable: entre 15% y 35%.</p> <p>Rango planeado: menor a 15%, hasta 4%.</p> <p>Rango excede: menor a 4%.</p>	<p>Es muy importante prevenir la corrupción de datos en los accesos a la aplicación CargoX, es por esto que el rango planeado debe ser menor a 15%, y optimo menor a 4%, tratando de evitar entrar en un rango mayor a esto. Dicho con otras palabras, cuanto más cerca se esta de 0%, menor es la cantidad de datos corruptos al ingresar al sistema.</p>
<i>Protocolos de autenticación</i>	<p>Rango no aceptable: menor a 60%.</p> <p>Rango aceptable: entre 60% y 85%.</p> <p>Rango planeado: mayor a 85%, hasta 98%.</p> <p>Rango excede: mayor a 98%.</p>	<p>Para esta métrica el rango no aceptable es menor a 60%, el aceptable entre 60% y 85%, el planeado va desde 86% a 98% (deseado) y óptimo mayor a 98%.</p> <p>Para obtener este porcentaje se tiene en cuenta la función <math>X = A / B</math> definida en la tabla 12 para esta métrica.</p>

Tabla 13 - Niveles de rating para evaluar los requerimientos de calidad

#### 6.4.- Criterios de evaluación

Los criterios de evaluación que se tendrán en cuenta de cada característica, subcaracterística y del resultado final son los categorizados en la sección 6.3.-*Niveles de rating*, siendo éstos los rangos no aceptable, aceptable, planeado y excede. Anteriormente se le asignó a cada métrica seleccionada un nivel de rating. Este nivel determinará el consecuente nivel de rating de las subcaracterísticas de calidad, las cuales determinarán el de las características de calidad, llegando, por último, el nivel de rating del producto de software (CargoX), asignándole formalmente su grado de calidad de acuerdo a lo evaluado. Es necesario, para esto, marcar cada uno de los rangos con un valor numérico, indicado en la siguiente tabla (tabla 14):

Rango del nivel de rating	Valores numéricos
NO ACEPTABLE	1
ACEPTABLE	2
PLANEADO	3
EXCEDE	4

Tabla 14 - Valores numéricos de cada nivel de rating

##### 6.4.1.- Criterios de evaluación de las subcaracterísticas de calidad

El criterio de evaluación de las subcaracterísticas depende directamente del resultado de las métricas.

Para determinar el nivel de rating de una subcaracterística se calcula el promedio de los valores numéricos de los niveles de rating obtenidos por las métricas. Al resultado que se obtiene se redondea hacia abajo, es decir, si por ejemplo se obtiene 1.5, el valor queda con 1 y según el resultado final obtenido se asigna el nivel de rating correspondiente según la tabla 14.

Se menciona el siguiente ejemplo:

Para la subcaracterística *accesibilidad* tomamos la métrica *número de idiomas soportados*. Supongamos que la primera tuvo un nivel de rating planeado, el cual posee valor numérico 3, y la segunda aceptable, con valor numérico 2. El promedio de estos dos valores nos da 2.5 ( $(3 + 2)/2 = 2.5$ ) y redondeando hacia abajo obtenemos el valor 2, el cual si observamos la tabla 14, nos da un nivel de rating planeado para ese valor obtenido para la subcaracterística *accesibilidad*.

##### 6.4.2.- Criterios de evaluación de las características de calidad

Una vez obtenidos los niveles de rating de cada subcaracterística, se calcula el promedio de los valores numéricos de cada una de ellas (véase correspondencia en la tabla 14) utilizando el mismo método descrito con anterioridad (redondeando hacia abajo), y, a partir de este promedio, se le asigna el nivel de rating correspondiente según la tabla 14 a la característica de calidad.

##### 6.4.3.- Criterios de evaluación final de la calidad

Para determinar el resultado final de la evaluación de calidad de la aplicación móvil CargoX se siguen los mismos pasos que se mencionaron en 6.4.1.- *Criterios de evaluación de las subcaracterísticas de calidad* y 6.4.2.- *Criterios de evaluación de las características de calidad*. Se toman los valores numéricos que se corresponden a los

niveles de rating obtenidos previamente de las dos características de calidad a evaluar (*usabilidad* y *seguridad*), se calcula el promedio (siempre redondeando hacia abajo), y, finalmente, se obtiene el nivel de rating final de la evaluación de calidad.

### 6.5.- Conclusiones

El alcance de la evaluación es la aplicación móvil CargoX, desarrollada para el sistema operativo Android. Se seleccionaron las métricas a aplicar para evaluar cada requerimiento de calidad. Estas métricas elegidas son externas, ya que son aquellas aplicables al software en ejecución.

Luego se definieron rangos de valores para cada métrica según su importancia (dada por los desarrolladores de CargoX), variando de un porcentaje mínimo a uno máximo. Estos rangos podrían variar en las siguientes iteraciones de evaluación de calidad a realizarse durante el proceso de mejora continua.

Se definieron también los niveles de rating posibles a los rangos no aceptable, aceptable, planeado y excede, dándole a cada uno un valor numérico, el cual define posteriormente el nivel de calidad de las subcaracterísticas de calidad, características de calidad y finalmente de la aplicación móvil.

El rango que se espera obtener de cada resultado de las métricas propuestas es el planeado, y, preferentemente, excede. Esto indica el grado de satisfacción pretendido de los requerimientos de calidad. Si gran parte de los resultados se encuentran en alguno de estos dos rangos, la aplicación se considera que es de muy buena calidad. Cada correspondencia de las métricas con su nivel de rating es considerada como un objetivo de calidad.

---

## CAPITULO 7.- PLAN DE EVALUACION

---

### 7.1.- Introducción

El diseño de la evaluación está basado en la descripción del producto a evaluar, en los requerimientos de la evaluación y en la especificación de la evaluación. Debe documentar los procedimientos que serán usados por el evaluador para realizar las mediciones determinadas en la especificación de evaluación y contener toda la información necesaria para llevar a cabo la misma. El resultado del diseño de la evaluación es el plan de evaluación.

En este capítulo se presenta el plan de evaluación de calidad del producto de software. Se describen los métodos, herramientas y procedimientos operativos necesarios para implementar la especificación de la evaluación.

#### 7.1.1.- Objetivo

El objetivo de la evaluación es obtener el nivel de calidad del producto de software identificando sus debilidades y fortalezas, con la finalidad de alimentar un proceso de mejora continua a través de la eliminación y corrección de sus debilidades. El desarrollo completo del objetivo se encuentra en la sección *5.1.- Propósito de la evaluación*, y la motivación de la evaluación se encuentra explicada en la sección *5.2.- Motivación de la evaluación*.

#### 7.1.2.- Descripción del producto a evaluar

El producto de software al cual se le va a realizar la evaluación de calidad es una aplicación móvil desarrollada para sistema operativo Android versión 4.0 en adelante, utilizable en cualquier dispositivo móvil que posea estas características. Brevemente, se encarga de ser un nexo entre transportistas y cargas a transportar por personas o empresas.

#### 7.1.3.- Requerimientos de la evaluación

Este punto fue detallado durante todo el *CAPITULO 5.- REQUISITOS DE LA EVALUACION*, donde se mencionó el propósito de la evaluación de acuerdo al uso del producto. Allí se describe el producto, el modelo de calidad seleccionado para la evaluación junto con las características de calidad, los requerimientos, grado de rigurosidad con la que debe realizarse la evaluación y grado de confidencialidad de los elementos involucrados en ella.

#### 7.1.4.- Alcance de la evaluación

Como bien se mencionó en la sección *6.1.- Alcance de la evaluación*, abarca la aplicación móvil CargoX, incluyendo la totalidad del sistema, exceptuando componentes que no son parte del mismo.

## 7.2.- Características de calidad aplicables

Las características de calidad que se aplican al presente trabajo son *usabilidad* y *seguridad*, con sus respectivas subcaracterísticas. Las mismas se mencionan y detallan en el capítulo 5, sección 5.4.1.- *Selección de características a evaluar*.

El modelo de calidad que se va a utilizar en la evaluación de calidad para el producto de software seleccionado se encuentra definido en la norma ISO/IEC 25010- *System and software quality models*, como bien se mencionó en la sección 5.4.- *Modelo de calidad*.

Los requerimientos de calidad definidos en 5.5.- *Requerimientos de calidad* apuntan a evaluar las características de calidad mencionadas y sus subcaracterísticas.

## 7.3.- Lista de prioridades

En la tabla 15 se muestran las características y subcaracterísticas a evaluar por orden de prioridad de forma descendente (a menor valor, mayor es la prioridad):

Característica	Subcaracterística	Nivel de prioridad
<i>Seguridad</i>	Prevención de corrupción de datos	1
<i>Seguridad</i>	Protocolos de autenticación	1
<i>Usabilidad</i>	Operabilidad	2
<i>Usabilidad</i>	Inteligibilidad	2
<i>Usabilidad</i>	Protección frente a errores de usuario	2
<i>Usabilidad</i>	Aprendizaje	3
<i>Usabilidad</i>	Accesibilidad	3
<i>Usabilidad</i>	Estética	4

**Tabla 15 - Listado de prioridades sobre las características y subcaracterísticas de calidad a evaluar**

En la tabla 15 se observa claramente las prioridades. En principio, la característica de *seguridad* con todas sus subcaracterísticas van a tener prioridad uno a la hora de ser evaluadas ya que es muy importante para una aplicación móvil proteger la información y los datos de manera que personas o sistemas no autorizados no puedan leerlos o modificarlos de manera fraudulenta.

Luego, para la característica de *usabilidad*, se definen prioridades que van de dos a cuatro. *Operabilidad*, *inteligibilidad* y *protección frente a errores de usuario* con prioridad dos, teniendo en cuenta que estas tres subcaracterísticas son muy importantes ya que definen el nivel de control de las operaciones a realizar. Más abajo con prioridad tres se encuentran *aprendizaje* y *accesibilidad*, y por último con prioridad cuatro la subcaracterística de *estética*. Estas últimas son muy importantes a pesar de la baja prioridad que posean. Indican la capacidad de la aplicación para enseñar al usuario a utilizarla, poseer contenido accesible para usuarios con discapacidades o

capacidades reducidas y que ese contenido sea agradable a la vista estéticamente hablando.

El grado de rigurosidad que se aplicará a la evaluación de calidad fue definido anteriormente en la sección 5.6.- *Grado de rigurosidad*.

#### 7.4.- Objetivos de calidad

Se definieron requerimientos de calidad en la sección 5.5.- *Requerimientos de calidad*, los cuales son medibles sobre el producto de software final con las métricas que fueron definidas en la sección 6.2.- *Selección de métricas* para las subcaracterísticas de las características de calidad seleccionadas, y en 6.3.- *Niveles de rating* se definieron los niveles de satisfacción de los requerimientos de calidad que puede alcanzar el resultado de cada métrica. Cada correspondencia de las métricas con su nivel de rating (tabla 13), es considerada como un objetivo de calidad del producto.

Si bien hay requerimientos de calidad que son medidos por más de una métrica, cada requerimiento pertenece a una subcaracterística y el resultado de las métricas de cada subcaracterística se combina para formar el resultado de la característica.

De acuerdo a lo definido en 6.4.- *Criterios de evaluación* sobre cómo combinar los resultados de las mediciones para obtener los resultados de calidad de las subcaracterísticas (sección 6.4.1.- *Criterios de evaluación de las subcaracterísticas de calidad*), de las características (sección 6.4.2.- *Criterios de evaluación de las características de calidad*) y del producto de software final (sección 6.4.3.- *Criterios de evaluación final de la calidad*), se tiene como objetivo de calidad primordial alcanzar resultados dentro del rango aceptable y, preferentemente, rango planeado o, mejor aún, rango excede, siendo este un objetivo de calidad que se desprende de los recientemente mencionados. Se desea obtener un producto que sea de muy buena calidad con respecto a las características seleccionadas, proveyendo al usuario de facilidad de uso y aprendizaje, acceso fácil, controlado y seguro, y una interfaz de usuario amigable y accesible.

#### 7.5.- Conclusiones

Una vez definido el plan de evaluación quedan documentados los procedimientos a utilizar por el evaluador para realizar las mediciones especificadas, describiendo los métodos, herramientas y procedimientos operativos necesarios para implementar la especificación de la evaluación.

Fueron especificados los objetivos de la evaluación, características de calidad ordenadas por prioridad (con sus subcaracterísticas), el alcance de la evaluación y los objetivos de calidad, así como también se realizó una breve descripción del producto de software a evaluar.

Un punto perseguido por la empresa, y por ende importante en la evaluación es lograr que tanto las subcaracterísticas, como las características y la aplicación móvil final obtengan un nivel de rating elevado, preferentemente PLANEADO o EXCEDE (a partir de las métricas que fueron seleccionadas), asegurando ésto que el producto de software provee la *seguridad* que necesita el usuario, así como también un nivel de *usabilidad* confiable de acuerdo a los requerimientos de la evaluación.

Con todas estas herramientas se inicia el proceso de ejecución de la evaluación de calidad, abordado en el *Capítulo 8*.



## CAPITULO 8.- EJECUCION DE LA EVALUACION

---

### 8.1.- Ejecución de la evaluación de la característica *seguridad*

En esta sección se llevará a cabo la ejecución de la evaluación de calidad para la característica de *seguridad* sobre la aplicación móvil CargoX. Las medidas de *seguridad* se utilizan para evaluar el grado en que un producto o sistema protege la información y los datos para que las personas u otros productos o sistemas tengan el grado de acceso a los datos apropiado para sus tipos y niveles de autorización.

Las subcaracterísticas a evaluar son: *integridad* y *autenticidad*.

Las métricas seleccionadas para realizar la evaluación de calidad sobre la característica de *seguridad* se encuentran definidas en la sección 6.2.2.- *Métricas seleccionadas para evaluar seguridad*.

#### 8.1.1.- Ejecución de la evaluación de la subcaracterística *integridad*

Se llevará a cabo la evaluación de calidad de la subcaracterística *integridad* sobre la aplicación móvil CargoX.

Como se mencionó en la sección 6.2.2, se utilizará la métrica *prevención de corrupción de datos* para evaluar esta subcaracterística.

##### 8.1.1.1.- Medición de la métrica *prevención de corrupción de datos*

Para este cálculo se realizará el inicio de sesión en reiteradas oportunidades con un usuario de la app contabilizando el número de accesos que corrompen los datos del mismo en un tiempo de observación estipulado en los primeros 15 días del mes de Febrero del 2018.

##### 8.1.1.1.1.- Cálculos de los elementos de datos de la métrica

En la tabla 16 se muestran los valores de los elementos de datos (QME's) que se definen en la tabla 12:

Métrica	<i>Prevención de corrupción de datos</i>	
Elementos de medida de calidad	Descripción	Valor
A	Número de accesos que realmente corrompen los datos.	0
B	Número de accesos donde se debe prevenir daño o rotura de datos.	150

**Tabla 16 - Elementos de datos para la métrica *Prevención de corrupción de datos***

Los valores de los elementos de datos de la tabla 16 se originan de:

- El tiempo de observación para esta métrica es de 15 días (primeros 15 días de Febrero del 2018).
- Se utiliza un usuario de prueba brindado por los desarrolladores de CargoX y se ingresa en reiteradas oportunidades a la app.
- 0 indica que es 0 el número de veces que los datos del usuario fueron corrompidos durante el tiempo de observación mencionado con anterioridad.
- 150 es la cantidad de accesos donde se deben prevenir daños o rotura de datos (10 accesos por día durante 15 días).

#### 8.1.1.1.2.- Cálculo de la métrica

La función que debe utilizarse para realizar el cálculo a la métrica *prevención de corrupción de datos* fue definida en la tabla 12. La fórmula es:  $X = A / B$ .

$$X = A / B = 0 / 150 = 0$$

#### 8.1.2- Ejecución de la evaluación de la subcaracterística autenticidad

Se llevará a cabo la evaluación de calidad de la subcaracterística *autenticidad* sobre la aplicación móvil CargoX.

##### 8.1.2.1- Medición de la métrica *protocolos de autenticación*

Se aplicará la métrica *protocolos de autenticación* con el propósito de evaluar qué tan bien autentica el sistema la identidad de un sujeto o recurso. Para esto se requiere el número de protocolos de autenticación (o campos requeridos para el login) que proporciona la app, y el número de los mismos que se encuentran en la especificación.

##### 8.1.2.1.1.- Cálculos de los elementos de datos de la métrica

En la tabla 17 se muestran los valores de los elementos de datos que se definen en la tabla 12:

Métrica	<i>Protocolos de autenticación</i>	
Elementos de medida de calidad	Descripción	Valor
A	Número de protocolos de autenticación proporcionados (p. Ej., ID de usuario / contraseña o tarjeta IC, etc).	3
B	Número de protocolos de autenticación requeridos en la especificación.	3

Tabla 17 - Elementos de datos para la métrica *protocolos de autenticación*

Los valores de los elementos de datos de la tabla 17 se originan de:

- 3 es el número de protocolos de autenticación que son visibles para el usuario dentro de la aplicación móvil CargoX (estos son: número de teléfono, código SMS y número de CPF).
- 3 es el número de protocolos de autenticación requeridos que se encuentran en la especificación (mencionados en el punto anterior). Se pueden observar en la sección 4.3.2.- *Pantallas del producto de software*, en las pantallas de login.

#### 8.1.2.1.2.- Cálculo de la métrica

La función que se utiliza para realizar el cálculo a la métrica *protocolos de autenticación* fue definida en la tabla 12. La fórmula es:  $X = A / B$ .

$$X = A / B = 3 / 3 = 1 = 100\%$$

### 8.2.- Ejecución de la evaluación de la característica *usabilidad*

En esta sección se llevará a cabo la ejecución de la evaluación de calidad para la característica de *usabilidad* sobre la aplicación móvil CargoX.

Los resultados se influyen por las capacidades de los usuarios y la configuración del sistema. Esto no invalida las mediciones realizadas debido a que el software de evaluación se ejecuta bajo condiciones explícitamente especificadas por una muestra de usuarios que representan a un grupo identificado. Las métricas de *usabilidad* arrojan, inevitablemente, resultados subjetivos.

Las subcaracterísticas a evaluar son: *operabilidad, inteligibilidad, protección frente a errores de usuario, aprendizaje, accesibilidad y estética*.

Las métricas seleccionadas para realizar la evaluación de calidad sobre la característica de *usabilidad* se encuentran definidas en la sección 6.2.1.- *Métricas seleccionadas para evaluar usabilidad*.

#### 8.2.1.- Ejecución de la evaluación de la subcaracterística *operabilidad*

Para realizar las medidas de *operabilidad* se utiliza un reporte brindado por los desarrolladores de CargoX, en el cual se encuentran las funciones descritas en la descripción del producto.

A continuación se presenta en la tabla 18 el reporte mencionado anteriormente.

Funciones	ID Reporte	Detalle
Login	0026	Se inicia sesión.
Aceptar pedido	0027	Se acepta el pedido de un cliente por parte de un usuario.
Rechazar pedido	0028	Se rechaza el pedido de un cliente por parte de un usuario.
Contactar cliente	0029	Se contacta al cliente por medio del envío de correo electrónico o llamada telefónica.

Iniciar transporte	0030	Se inicia el envío de la carga junto con el trayecto a recorrer.
Cancelar transporte	0031	Se cancela el transporte de la carga.
Seguimiento del transporte (trackeo)	0032	Se realiza el seguimiento del transporte de la carga.
Listado de cargas pendientes a transportar	0033	Se listan las cargas pendientes a transportar.
Visualizar detalles de carga	0034	Se visualizan los detalles de la carga.
Modificación del estado del transporte de la carga	0035	Se modifica el estado del transporte de la carga.
Listado de cargas ya transportadas (histórico de cargas)	0036	Se lista el historial de cargas ya transportadas por el usuario.

Tabla 18 - Reporte de funciones descritas en la descripción del producto

#### 8.2.1.1.- Medición de la métrica *posibilidad de personalización*

Se aplicará la métrica *posibilidad de personalización* con el propósito de evaluar la porción de funciones y procedimientos operativos que puede ser personalizada por los usuarios según lo deseen. Para esto se requiere conocer el número total de funciones y procedimientos operativos implementados en CargoX y aquellos que pueden ser personalizados a conveniencia del usuario. La totalidad de las funciones se encuentra detallada en la tabla 18.

##### 8.2.1.1.1.- Cálculos de los elementos de datos

En la tabla 19 se muestran los valores de los elementos de datos que se definen en la tabla 12:

Métrica	<i>Posibilidad de personalización</i>	
Elementos de datos	Descripción	Valor
A	Número de funciones y procedimientos operativos que pueden personalizarse para la comodidad del usuario.	0
B	Número total de funciones y procedimientos operativos implementados.	11

Tabla 19 - Elementos de datos para la métrica *posibilidad de personalización*

Los valores de los elementos de datos de la tabla 19 se originan de:

- La tabla 18 - Reporte de funciones descritas en la descripción del producto.
- 11 es el número total de funciones operativas implementadas por CargoX.
- 0 es el número de funciones que pueden ser personalizadas por los usuarios dentro de la aplicación móvil. Este valor se obtuvo mediante la utilización de un usuario de prueba y observando la sección 4.3.2.- *Pantallas del producto de software*.

#### 8.2.1.1.2.- Cálculo de la métrica

La función que debe utilizarse para realizar el cálculo a la métrica *posibilidad de personalización* fue definida en la tabla 12. La fórmula es:  $X = A / B$ .

$$X = A / B = 0 / 11 = 0$$

#### 8.2.1.2.- Medición de la métrica *claridad del mensaje*

Se aplica la métrica con el propósito de evaluar aquellos mensajes dentro de la aplicación móvil CargoX que se pueden entender con facilidad. Para esto se necesita conocer el número de mensajes que son entendidos por los usuarios, así como el número total de mensajes implementados en la aplicación móvil.

##### 8.2.1.2.1.- Cálculos de los elementos de datos de la métrica

En la tabla 20 se muestran los valores de los elementos de datos que se definen en la tabla 12:

Métrica	Claridad del mensaje	
Elementos de datos	Descripción	Valor
A	Número de mensajes que son entendidos por los usuarios simulados.	25
B	Número total de mensajes implementados.	29

**Tabla 20 - Elementos de datos para la métrica *claridad del mensaje***

Los valores de los elementos de datos de la tabla 20 se originan de:

- 25 es el número de mensajes entendibles.
- 29 es el número total de mensajes implementados (tanto entendibles como no entendibles).
- Parte de estos valores fueron obtenidos a partir del ingreso a la aplicación móvil de CargoX con un usuario de prueba (en una versión beta). Pueden observarse con más certeza estos valores en la sección 4.3.2.- *Pantallas del producto de software*.
- Se toma como mensaje no sólo las alertas y mensajes de error, sino cualquier mensaje que explique al usuario los pasos a seguir o qué debe hacer a continuación.

### 8.2.1.2.2.- Cálculo de la métrica

La función que debe utilizarse para realizar el cálculo a la métrica *claridad del mensaje* fue definida en la tabla 12. La fórmula es:  $X = A / B$ .

$$X = A / B = 25 / 29 \cong 0,86206896 \cong 0,86$$

### 8.2.2.- Ejecución de la evaluación de la subcaracterística inteligibilidad

Se lleva a cabo la evaluación de calidad de la subcaracterística *inteligibilidad* sobre la aplicación móvil CargoX. Las medidas de *inteligibilidad* apropiadas se pueden usar para evaluar si los nuevos usuarios pueden comprender:

- Si el producto o sistema de software es adecuado para sus propósitos o no.
- Cómo se puede usar para tareas particulares.

Como se mencionó en la sección 6.2.1, se utilizará la métrica *capacidad de demostración* para evaluar esta subcaracterística.

#### 8.2.2.1.- Medición de la métrica *capacidad de demostración*

Se aplicará la métrica *capacidad de demostración* con el propósito de evaluar la proporción de funciones que tienen la capacidad de demostración frente a usuarios para que los mismos reconozcan su idoneidad. Para esto se necesita el número total de funciones que existen en la descripción del producto (extraídas de la tabla 18 - Reporte de funciones descritas en la descripción del producto) y el número de funciones implementadas que poseen capacidad de demostración.

##### 8.2.2.1.1.- Cálculos de los elementos de datos de la métrica

En la tabla 21 se muestran los valores de los elementos de datos que se definen en la tabla 12:

Métrica	Capacidad de demostración	
Elementos de datos	Descripción	Valor
A	Número de funciones implementadas con capacidad de demostración.	11
B	Número total de funciones en la descripción del producto.	11

**Tabla 21 - Elementos de datos para la métrica *capacidad de demostración***

Los valores de los elementos de datos de la tabla 21 se originan de:

- 11 es el número total de funciones descritas en la descripción del producto, extraídas de la tabla 18 Reporte de funciones; y también es el número de funciones implementadas con capacidad de demostración. Para obtener este valor se utilizó un usuario de prueba para comprobar si cada una de las funcionalidades que nos brinda la app nos permite entender si el producto es el adecuado para nuestras necesidades. Se pueden observar las funciones con capacidad de

demostración más en detalle en la sección 4.3.2.- *Pantallas del producto de software*.

#### 8.2.2.1.2.- Cálculo de la métrica

La función que debe utilizarse para realizar el cálculo a la métrica *capacidad de demostración* fue definida en la tabla 12. La fórmula es:  $X = A / B$ .

$$X = A / B = 11 / 11 = 1$$

#### 8.2.3.- Ejecución de la evaluación de la subcaracterística *protección frente a errores de usuario*

Como se mencionó en la sección 6.2.1, se utilizará la métrica *comprobación de errores de entrada de usuario* para evaluar esta subcaracterística.

La tabla 22 muestra los errores ocasionados (de forma intencional o no) por un usuario de prueba brindado por los desarrolladores de la aplicación móvil CargoX durante los primeros 15 días del mes de Febrero del 2018, identificándolos por errores de entrada del usuario y otro tipo de error, teniendo en cuenta que para la medición de la métrica *comprobación de errores de entrada de usuario* sólo nos enfocaremos en aquellos errores que sean de entrada.

Fecha	Descripción	¿Es error de entrada del usuario?	Detalle
03/02/2018	Error al llamar telefónicamente a los operadores de la aplicación.	No	Cuando el usuario desea comunicarse telefónicamente con los operadores por x razón, se muestra cartel de error especificando claramente el motivo y no se realiza la llamada.
04/02/2018	Error en la visualización del perfil de usuario.	No	Al querer ingresar al perfil de usuario la aplicación re direcciona a historial de cargas.
07/02/2018	Error en el ingreso de usuario y/o contraseña.	Sí	Al querer iniciar sesión, el usuario ingresa los datos de entrada pero estos no son los correctos. Se muestra un mensaje claro de error.
10/02/2018	Error al completar el formulario.	Si	El usuario no completó adecuadamente los campos de entrada. Se indica mediante un mensaje de alerta claro y comprensible para el mismo.



12/02/2018	Error en el ingreso de número de teléfono y/o número de CPF.	Si	El sistema no permite al usuario ingresar la contraseña, imposibilitándolo de iniciar sesión en la aplicación. Avisa con un alerta clara.
------------	--	----	---

Tabla 22 - Errores de usuario

#### 8.2.3.1.- Medición de la métrica *comprobación de errores de entrada de usuario*

Para evaluar la proporción de errores detectables de entrada del usuario identificada por el sistema (y claramente descrita para facilitar la corrección del/los error/es), se requiere la totalidad de errores de entrada de usuario detectables, así como también el número de errores identificados y claramente descritos. Se utilizará la tabla 22 para obtener los datos requeridos y así realizar los cálculos para esta métrica.

##### 8.2.3.1.1.- Cálculos de los elementos de datos de la métrica

En la tabla 23 se muestran los valores de los elementos de datos que se definen en la tabla 12:

Métrica	<i>Comprobación de errores de entrada de usuario</i>	
Elementos de datos	Descripción	Valor
A	Número de errores identificados y claramente descritos para corregir los errores de entrada del usuario por el sistema.	3
B	Número total de errores detectables de entrada de usuario.	3

Tabla 23 - Elementos de datos para la métrica *comprobación de errores de entrada de usuario*

Los valores de los elementos de datos de la tabla 23 se originan de:

- 3 es el número de errores de entrada de usuario identificados que son claramente descriptos mediante una alerta al usuario extraídos de la tabla 22.
- 3 es el número total de errores de entrada de usuario detectables extraídos de la tabla 22.

##### 8.2.3.1.2.- Cálculo de la métrica

La función que debe utilizarse para realizar el cálculo a la métrica *comprobación de errores de entrada de usuario* fue definida en la tabla 12. La fórmula es:  $X = A / B$ .

$$X = A / B = 3 / 3 = 1$$

#### 8.2.4.- Ejecución de la evaluación de la subcaracterística aprendizaje

Se llevará a cabo la evaluación de calidad de la subcaracterística *aprendizaje* sobre la aplicación móvil CargoX.

Como se mencionó en la sección 6.2.1, se utilizará la métrica *campos de entrada con valores predeterminados* para evaluar esta subcaracterística.

##### 8.2.4.1.- Medición de la métrica *campos de entrada con valores predeterminados*

Se aplicará la métrica *campos de entrada con valores predeterminados* con el propósito de evaluar la proporción de campos de entrada con valores predeterminados que sean útiles y puedan rellenarse automáticamente. Para esto se requiere la totalidad de campos de entrada existentes en la aplicación móvil, así como también el número de campos de entrada con valores útiles por defecto.

##### 8.2.4.1.1.- Cálculos de los elementos de datos de la métrica

En la tabla 24 se muestran los valores de los elementos de datos que se definen en la tabla 12:

Métrica	<i>Campos de entrada con valores predeterminados</i>	
Elementos de datos	Descripción	Valor
A	Número de campos de entrada que podrían tener valores por defecto útiles.	9
B	Número de campos de entrada total.	9

**Tabla 24 - Elementos de datos para la métrica *campos de entrada con valores predeterminados***

Los valores de los elementos de datos de la tabla 24 se originan de:

- 9 es el número total de campos de entrada a rellenar por el usuario que posee la aplicación móvil CargoX, así como también es el número de estos campos de entrada que poseen valores por defecto útiles para poder ser rellenados. Para obtener este valor se observó la sección 4.3.2.- *Pantallas del producto de software*.

##### 8.2.4.1.2.- Cálculo de la métrica

La función que debe utilizarse para realizar el cálculo a la métrica *campos de entrada con valores predeterminados* fue definida en la tabla 12. La fórmula es:  $X = A / B$ .

$$X = A / B = 9 / 9 = 1$$

### 8.2.5.- Ejecución de la evaluación de la subcaracterística accesibilidad

Las medidas de *accesibilidad* se utilizan para evaluar el grado en que un producto o sistema puede ser utilizado por personas con la más amplia gama de características y capacidades para lograr un objetivo específico en un contexto específico de uso.

Como se mencionó en la sección 6.2.1, se utilizará la métrica *número de idiomas soportados* para evaluar esta subcaracterística.

#### 8.2.5.1.- Medición de la métrica *número de idiomas soportados*

Se requiere el número necesario de idiomas a ser soportados que se encuentran definidos en la descripción del producto, así como también la cantidad de idiomas que realmente soporta la aplicación.

##### 8.2.5.1.1.- Cálculos de los elementos de datos de la métrica

En la tabla 25 se muestran los valores de los elementos de datos que se definen en la tabla 12:

Métrica	Número de idiomas soportados	
Elementos de datos	Descripción	Valor
A	Número de idiomas realmente soportados.	1
B	Número necesario de idiomas a ser soportados definidos en la descripción del producto.	1

**Tabla 25 - Elementos de datos para la métrica *número de idiomas soportados***

Los valores de los elementos de datos de la tabla 25 se originan de:

- 1 es el número de idiomas que se indica en la descripción del producto que son soportados por la aplicación móvil. Este dato fue brindado por los desarrolladores de CargoX.
- 1 es el número de idiomas que realmente soporta la app.

Observación: el idioma soportado definido en la descripción del producto es Portugués, y también es el idioma que realmente soporta CargoX en su totalidad. Esto fue consultado con los desarrolladores de CargoX y observado mediante la utilización de un usuario de prueba.

##### 8.2.5.1.2.- Cálculo de la métrica

La función que debe utilizarse para realizar el cálculo a la métrica *número de idiomas soportados* fue definida en la tabla 12. La fórmula es:  $X = A / B$ .

$$X = A / B = 1 / 1 = 1$$

### 8.2.6.- Ejecución de la evaluación de la subcaracterística estética

Como se mencionó en la sección 6.2.1, se utilizará la métrica *personalización de los elementos de la interfaz de usuario* para evaluar esta subcaracterística.

#### 8.2.6.1.- Medición de la métrica *personalización de los elementos de la interfaz de usuario*

Para evaluar la proporción de elementos de la interfaz de usuario que pueden ser personalizados en apariencia se requiere la totalidad de elementos que posee la interfaz de usuario, así como también aquellos elementos dentro de la totalidad que pueden ser personalizados a conveniencia por el usuario.

##### 8.2.6.1.1.- Cálculos de los elementos de datos de la métrica

En la tabla 26 se muestran los valores de los elementos de datos (QME's) que se definen en la tabla 12:

Métrica	<i>Personalización de los elementos de la interfaz de usuario</i>	
Elementos de datos	Descripción	Valor
A	Número de elementos de la interfaz de usuario que se pueden personalizar.	0
B	Número total de elementos de la interfaz de usuario.	42

**Tabla 26 - Elementos de datos para la métrica *personalización de los elementos de la interfaz de usuario***

Los valores de los elementos de datos de la tabla 26 se originan de:

- 0 es el número de elementos de la interfaz de usuario dentro de la aplicación móvil CargoX que pueden ser personalizados por los usuarios.
- 42 es el número total de elementos de la interfaz de usuario que posee la app. Estos pueden ser animaciones, iconos, figuras, botones, fondos, menús, etc.
- Estos valores fueron extraídos utilizando un usuario de prueba y observando la sección 4.3.2.- *Pantallas del producto de software*.

##### 8.2.6.1.2.- Cálculo de la métrica

La función que debe utilizarse para realizar el cálculo a la métrica *personalización de los elementos de la interfaz de usuario* fue definida en la tabla 12. La fórmula es:  $X = A / B$ .

$$X = A / B = 0 / 42 = 0$$

### 8.3.- Conclusiones

Se ha documentado la evaluación de calidad sobre la aplicación móvil CargoX. Para esto se extrajo la información de tablas de reportes realizados por la empresa Sontra para esta aplicación, las cuales fueron brindadas para el presente trabajo, y de la visualización de la aplicación móvil en sí.

Las métricas que fueron seleccionadas para ser aplicadas se tomaron del CAPITULO 6.- ESPECIFICACION DE LA EVALUACION donde fueron definidas, descritas y bien detalladas para realizar las mediciones de calidad, logrando obtener los resultados a través de cálculos realizados a partir de las fórmulas de cada métrica que se mencionan en el capítulo 6.

Por último, gracias a los cálculos realizados, se disponen de los resultados de las mediciones para proceder a la determinación de la calidad del producto de software en el siguiente capítulo.

---

## CAPITULO 9.- REPORTE DE LA EVALUACION

---

### 9.1.- Identificaciones

En cuanto a las identificaciones relativas a la evaluación realizada, tanto el proveedor del producto como el solicitante de la evaluación de calidad son la misma entidad, mientras que el evaluador del producto de software en cuestión no lo es.

La información del proveedor/solicitante del producto de software que puede darse a conocer es la descrita en el *CAPITULO 4*. La persona responsable de realizar la evaluación de calidad sobre la aplicación móvil CargoX (evaluador del producto de software) es el autor de este trabajo: *Ordóñez Pablo Andrés*, cuya evaluación es llevada a cabo en su respectivo domicilio.

#### 9.1.1.- Reporte de evaluación

En la tabla 27 se puede observar la identificación del presente reporte de evaluación:

Identificación del reporte de evaluación	
Número de serie	REV_CX_2018/02
Número de páginas	12
Fecha	01/03/2018
Producto evaluado	CargoX
Versión del producto evaluado	v.1.2.1

**Tabla 27 – Identificación del reporte de evaluación**

### 9.2.- Requerimientos de la evaluación

Los requerimientos de la evaluación de calidad fueron especificados en detalle durante todo el capítulo 5: *CAPITULO 5*. En esta sección se mencionará de forma breve cada uno de los mismos.

El producto de software seleccionado es una aplicación móvil desarrollada para Android versión 4.0, llamada CargoX. Es un producto terminado dirigido al consumidor final. La evaluación de calidad se realizará sobre la versión 1.2.1 del mismo.

La descripción completa del producto evaluado se encuentra en el capítulo 4: *CAPITULO 4*.

El propósito de la evaluación de calidad es obtener el nivel de calidad del mismo del producto de software para poder identificar sus debilidades y fortalezas de una manera formalmente medible, en el marco de un proceso de mejora continua.

Los requerimientos de calidad evaluados sobre el producto se encuentran especificados en la tabla 8, relacionando cada uno con la característica y la subcaracterística de calidad a la cual pertenece. Luego, en la tabla 9 se determina para cada requerimiento, el grado de rigurosidad requerido.

### 9.3.- Especificación de la evaluación

Todo lo referido a la especificación de la evaluación de calidad sobre el producto de software seleccionado se encuentra definido en el capítulo 6: *CAPITULO 6*. En esta sección se mencionará de forma breve una parte de la misma.

El alcance de la evaluación de calidad abarca el producto de software CargoX, el cuál es una aplicación móvil desarrollada para Android 4.0 en adelante, incluyendo la totalidad del sistema, exceptuando componentes que no son parte del mismo, como por ejemplo las API's a las cuales se conecta o el motor de base de datos que utiliza, o toda la parte del sistema que se ejecuta mediante la utilización del dashboard.

En la sección 6.2.- *Selección de métricas* se definieron la especificación de mediciones realizadas sobre el producto y la correspondencia entre estas y los requerimientos de calidad.

### 9.4.- Métodos de evaluación

Los métodos de evaluación consisten en las métricas que se aplicarán y los detalles de su aplicación y evaluación. En la sección 6.2.- *Selección de métricas* se observan las métricas que fueron seleccionadas para cada subcaracterística de calidad dentro de las características escogidas.

Cada métrica requiere de la obtención de una serie de elementos de datos para efectuar su cálculo. En la tabla 12 pueden observarse los elementos de datos, explicando el significado de cada uno. En esa misma tabla también se puede observar la función que debe ser utilizada para, a través de la manipulación de los elementos de datos (reemplazándolos con las mediciones efectuadas), obtener los resultados de cada métrica.

Gracias a la tabla 13 y a los criterios de evaluación a utilizar descritos en la sección 6.4.- *Criterios de evaluación* se puede comparar el resultado obtenido en los cálculos y asignarle un rango adecuado a cada métrica para evaluar así el grado de satisfacción con el requerimiento de calidad correspondiente a partir de los criterios de evaluación antes mencionados y así llegar a una conclusión final sobre la evaluación de calidad sobre la aplicación móvil CargoX.

### 9.5.- Resultados de la evaluación

Se informan los resultados de las métricas aplicadas en la evaluación así como los resultados de las subcaracterísticas y características de calidad, acompañados cada uno de una observación o comentario de evaluación y cerrando con la conclusión final de la evaluación de calidad del producto.

#### 9.5.1.- Resultado de la evaluación de las métricas de calidad

En la tabla 28 se informan los resultados de las métricas aplicadas en la evaluación de calidad.



Característica	Seguridad		Subcaracterística		Integridad
Métrica (X)	Interpretación	Rango / Valor de aceptación	Rango / Valor planeado	Resultado	Nivel de rating
<i>Prevención de corrupción de datos</i>	$0 \leq X \leq 1$ Cercano a 0 es mejor.	$0,15 < X \leq 0,35$ , es decir, la cantidad de datos corruptos a prevenir debe ser menor o igual que 35% y mayor a 15%.	$0,04 \leq X \leq 0,15$ , es decir, la cantidad de datos corruptos a prevenir debe ser menor o igual que 15% y mayor o igual a 4%.	X = 0, es decir, hay 0% de corrupción de datos.	EXCEDE
Característica	Seguridad		Subcaracterística		Autenticidad
Métrica (X)	Interpretación	Rango / Valor de aceptación	Rango / Valor planeado	Resultado	Nivel de rating
<i>Protocolos de autenticación</i>	$0 \leq X \leq 1$ Cercano a 1 es mejor.	$0,60 \leq X \leq 0,85$ , es decir, la cantidad de protocolos de autenticación proporcionados debe ser mayor o igual que 60% y menor o igual a 85%.	$0,85 < X \leq 0,98$ , es decir, la cantidad de protocolos de autenticación proporcionados debe ser mayor que 85% y menor o igual a 98%.	X = 1, es decir, el 100% de los protocolos de autenticación son proporcionados por CargoX.	EXCEDE
Característica	Usabilidad		Subcaracterística		Operabilidad
Métrica (X)	Interpretación	Rango / Valor de aceptación	Rango / Valor planeado	Resultado	Nivel de rating

<i>Posibilidad de personalización</i>	0 <= X <= 1 Cercano a 1 es mejor.	0,50 <= X <= 0,75, es decir, la cantidad de funciones de la aplicación móvil con capacidad de personalización debe estar entre el 50% inclusive y el 75% inclusive.	0,75 < X <= 0,95, es decir, la cantidad de funciones de la aplicación móvil con capacidad de personalización debe estar entre el 75% y el 95% inclusive.	X = 0, es decir, el 0% de las funciones operativas pueden ser personalizadas por los usuarios de CargoX a conveniencia.	NO ACEPTABLE
<i>Claridad del mensaje</i>	0 <= X <= 1 Cercano a 1 es mejor.	0,50 <= X <= 0,75, es decir, la cantidad de mensajes que pueden ser entendidos fácilmente debe estar entre el 50% inclusive y el 75% inclusive.	0,75 < X <= 0,95, es decir, la cantidad de mensajes que pueden ser entendidos fácilmente debe estar entre el 75% y el 95% inclusive.	X = 0,86, es decir, el 86% de los mensajes de la aplicación móvil CargoX pueden ser entendidos fácilmente.	PLANEADO
<b>Característica</b>	<b>Usabilidad</b>		<b>Subcaracterística</b>		<b>Inteligibilidad</b>
<b>Métrica (X)</b>	<b>Interpretación</b>	<b>Rango / Valor de aceptación</b>	<b>Rango / Valor planeado</b>	<b>Resultado</b>	<b>Nivel de rating</b>
<i>Capacidad de demostración</i>	0 <= X <= 1 Cercano a 1 es mejor.	0,50 <= X <= 0,75, es decir, la cantidad de funciones de la aplicación móvil con capacidad de demostración debe estar entre el 50% inclusive y el 75% inclusive.	0,75 < X <= 0,95, es decir, la cantidad de funciones de la aplicación móvil con capacidad de demostración debe estar entre el 75% y el 95% inclusive.	X = 1, es decir, el 100% de las funciones de la aplicación móvil CargoX poseen capacidad de demostración.	EXCEDE

Característica	Usabilidad		Subcaracterística		Protección frente a errores de usuario
Métrica (X)	Interpretación	Rango / Valor de aceptación	Rango / Valor planeado	Resultado	Nivel de rating
Comprobación de errores de entrada de usuario	0 <= X <= 1 Cercano a 1 es mejor.	0,65 <= X <= 0,85, es decir, la cantidad de errores identificados por el sistema y claramente descriptos debe estar entre el 65% inclusive y el 85% inclusive.	0,85 < X <= 0,98, es decir, la cantidad de errores identificados por el sistema y claramente descriptos debe estar entre el 85% y el 98% inclusive.	X = 1, es decir, el 100% de los errores de la aplicación móvil CargoX son identificados por el sistema y claramente descriptos para facilitar la corrección del error.	EXCEDE

Característica	Usabilidad		Subcaracterística		Aprendizaje
Métrica (X)	Interpretación	Rango / Valor de aceptación	Rango / Valor planeado	Resultado	Nivel de rating
Campos de entrada con valores predeterminados	0 <= X <= 1 Cercano a 1 es mejor.	0,50 <= X <= 0,75, es decir, la cantidad de campos de entrada que poseen valores por defecto útiles debe estar entre el 50% inclusive y el 75% inclusive.	0,75 < X <= 0,95, es decir, la cantidad de campos de entrada que poseen valores por defecto útiles debe estar entre el 75% y el 95% inclusive.	X = 1, es decir, el 100% de los campos de entrada de la aplicación móvil CargoX poseen valores por defecto que son útiles para el usuario.	EXCEDE

Característica	Usabilidad		Subcaracterística		Accesibilidad
Métrica (X)	Interpretación	Rango / Valor de aceptación	Rango / Valor planeado	Resultado	Nivel de rating

<i>Número de idiomas soportados</i>	0 <= X <= 1 Cercano a 1 es mejor.	0,50 <= X <= 0,75, es decir, la cantidad de idiomas soportados debe estar entre el 50% inclusive y el 75% inclusive.	0,75 < X <= 0,95, es decir, la cantidad de idiomas soportados debe estar entre el 75% y el 95% inclusive.	X = 1, es decir, el 100% de los idiomas que se encuentran en la descripción del producto son soportados por la aplicación móvil.	EXCEDE
<b>Característica</b>	<b>Usabilidad</b>		<b>Subcaracterística</b>		<b>Estética</b>
<b>Métrica (X)</b>	<b>Interpretación</b>	<b>Rango / Valor de aceptación</b>	<b>Rango / Valor planeado</b>	<b>Resultado</b>	<b>Nivel de rating</b>
<i>Personalización de los elementos de la interfaz de usuario</i>	0 <= X <= 1 Cercano a 1 es mejor.	0,50 <= X <= 0,75, es decir, la cantidad de elementos que pueden ser personalizados por los usuarios debe estar entre el 50% inclusive y el 75% inclusive.	0,75 < X <= 0,95, es decir, la cantidad de elementos que pueden ser personalizados por los usuarios debe estar entre el 75% y el 95% inclusive.	X = 0, es decir, el 0% de los elementos de la interfaz de usuario de la aplicación móvil CargoX pueden personalizarse por sus usuarios.	NO ACEPTABLE

Tabla 28 – Resultados de la evaluación de las métricas

En la tabla 29 se informan los comentarios y observaciones de los resultados de cada métrica.

<b>Métrica</b>	<b>Observaciones de la evaluación</b>
<i>Prevención de corrupción de datos</i>	El resultado de la evaluación demuestra que hay 0% de corrupción de datos. El rango esperado era PLANEADO y se obtuvo EXCEDE. Aun así es mejor el resultado.
<i>Protocolos de autenticación</i>	El resultado de la evaluación demuestra que todos los protocolos de autenticación son proporcionados por la aplicación móvil, obteniendo un rango EXCEDE, el cual es mejor al que se esperaba.

<i>Posibilidad de personalización</i>	De la totalidad de funciones operativas que posee la aplicación, el 0%, es decir, ninguna puede ser personalizada por usuarios de la misma. Este resultado entra dentro del rango NO ACEPTABLE, siendo esto muy desfavorable.
<i>Claridad del mensaje</i>	La mayoría de los mensajes que posee la aplicación móvil evaluada (el 86%) pueden ser entendidos fácilmente por sus usuarios, obteniendo con este resultado el rango esperado (PLANEADO).
<i>Capacidad de demostración</i>	La totalidad de las funciones de CargoX poseen capacidad de demostración. Este resultado se encuentra dentro del rango EXCEDE, siendo aún mejor que el rango esperado.
<i>Comprobación de errores de entrada de usuario</i>	La totalidad de los errores de CargoX son identificados por el sistema y descriptos claramente. El resultado de la evaluación se encuentra dentro del rango EXCEDE. Se esperaba alcanzar el rango PLANEADO por lo q se mejoró lo esperado.
<i>Campos de entrada con valores predeterminados</i>	Todos los campos de entrada que se encuentran en la aplicación móvil poseen valores por defecto que son útiles para el usuario. El resultado se encuentra en el rango EXCEDE.
<i>Número de idiomas soportados</i>	La cantidad total de idiomas que se encuentran en la descripción del producto son soportados por la aplicación. Nuevamente el resultado se encuentra en el rango EXCEDE, teniendo en cuenta que es mejor que el esperado, el cual era PLANEADO.
<i>Personalización de los elementos de la interfaz de usuario</i>	Ningún elemento de la interfaz de usuario de CargoX puede personalizarse por sus usuarios. Esto arroja un resultado que se encuentra en el rango NO ACEPTABLE, siendo el esperado el rango PLANEADO.

Tabla 29 – Observaciones de la evaluación de las métricas

### 9.5.2.- Resultados de la evaluación de las subcaracterísticas de calidad

En la tabla 30 a continuación, se informan los resultados de la evaluación de las subcaracterísticas de calidad y los comentarios y observaciones de los mismos. Para determinar el rating de cada subcaracterística se procede según lo definido en la sección 6.4.1.- *Criterios de evaluación de las subcaracterísticas de calidad*.

Subcaracterística	Característica	Nivel de rating	Observación de la evaluación
Integridad	<i>Seguridad</i>	EXCEDE	Se concluye que el producto de software evaluado consta de un nivel o grado de integridad muy alto, impidiendo accesos no autorizados o modificación de datos.
Autenticidad	<i>Seguridad</i>	EXCEDE	Se concluye que el producto de software evaluado provee de un alto grado de autenticidad, demostrando que la identidad de un sujeto o recurso es la que se reivindica.
Operabilidad	<i>Usabilidad</i>	ACEPTABLE	Se concluye que el producto de software evaluado permite al usuario operarlo y controlarlo.
Inteligibilidad	<i>Usabilidad</i>	EXCEDE	Se concluye que el producto de software evaluado posee la capacidad de permitir al usuario entender si la aplicación es adecuada para sus necesidades.
Protección frente a errores de usuario	<i>Usabilidad</i>	EXCEDE	Se concluye que el producto de software evaluado posee la capacidad para proteger a los usuarios de hacer errores.

Aprendizaje	<i>Usabilidad</i>	EXCEDE	Se concluye que el producto de software evaluado permite al usuario aprender de su uso de una manera óptima, es decir, les permite alcanzar los objetivos específicos la primera vez que utilizan la app.
Accesibilidad	<i>Usabilidad</i>	EXCEDE	Se concluye que el producto de software evaluado permite de manera óptima que sea utilizado por usuarios con determinadas características y discapacidades. Hay que tener en cuenta que sólo se tomó la métrica <i>número de idiomas soportados</i> para esta evaluación, ya que las otras dos métricas ( <i>accesibilidad para usuarios con discapacidad cognitiva o física</i> y <i>accesibilidad para usuarios con discapacidad auditiva o visual</i> ) no aplican para este tipo de aplicación móvil, que se trata sólo de usuarios transportistas los cuales no pueden tener este tipo de discapacidades a la hora de conducir un vehículo (por lo tanto el número de usuarios con estas discapacidades sería nulo).



Estética	Usabilidad	NO ACEPTABLE	Se concluye que el producto de software evaluado no posee la capacidad de agradar y satisfacer al usuario desde el punto de vista de su interfaz de usuario.
----------	------------	--------------	--

Tabla 30 - Observaciones de la evaluación de las subcaracterísticas

### 9.5.3.- Resultados de la evaluación de las características de calidad

En la tabla 31 a continuación, se informan los resultados de la evaluación de las características de calidad y los comentarios y observaciones de los mismos. Para determinar el rating de cada característica se procede según lo definido en la sección 6.4.2.- *Criterios de evaluación de las características de calidad*.

Característica	Nivel de rating	Observaciones de la evaluación
Seguridad	EXCEDE	Como se mencionó en la sección 5.4.1.2.- <i>Importancia de la evaluación de la característica seguridad para aplicaciones móviles</i> una de las características más importantes a la hora de hablar de aplicaciones móviles es la seguridad. En esta evaluación se concluye que la aplicación móvil CargoX mantiene de manera eficiente y óptima la integridad de los datos y verificación de que el usuario es quien dice ser.
Usabilidad	PLANEADO	Otra de las características más importantes en el ámbito de aplicaciones móviles es la de <i>usabilidad</i> , como bien se explica en la sección 5.4.1.1.- <i>Importancia de la evaluación de la característica usabilidad para aplicaciones móviles</i> . Se considera que el producto de software tiene la facilidad de que los usuarios puedan utilizarla para sus necesidades. De todos modos se recomienda poner énfasis en mejorar: <ul style="list-style-type: none"> <li>• Estética: para poder mejorar la interfaz de usuario de manera de agradar y satisfacer a los mismos.</li> <li>• Operabilidad: para poder mejorar la forma en que el producto permite al usuario operarlo y controlarlo.</li> </ul>

Tabla 31 - Observaciones de la evaluación de las características

#### 9.5.4.- Conclusión final acerca de la calidad del producto de software

Como conclusión de la evaluación de calidad realizada al producto de software CargoX, en base a lo definido en la sección 6.4.3.- *Criterios de evaluación final de la calidad*, se determina que el mismo alcanza el grado de calidad o nivel de rating PLANEADO en cuanto a *seguridad* y a *usabilidad*, lográndose la mayoría de los objetivos de calidad planteados en la sección 7.4.- *Objetivos de calidad* del plan de evaluación.

Se puede concluir, también, que se alcanzaron los resultados planificados para la mayoría de las métricas a evaluar, y para algunas de ellas se excedieron, siendo esto aún más beneficioso en cuanto a la calidad del producto evaluado. Este posee un nivel de seguridad muy confiable, y un nivel apropiado y totalmente aceptable de *usabilidad*.

Fortalezas del producto de software:

- Capacidad de brindar al usuario seguridad a la hora de autenticarse en el sistema.
- Capacidad de brindar al usuario seguridad para con sus datos personales, evitando que terceros puedan manipularlos.
- Capacidad de permitir al usuario operar y controlar las distintas funcionalidades de manera aceptable.
- Capacidad de permitir al usuario entender si es adecuada para sus necesidades.
- Posee la capacidad de proteger a los usuarios de cometer errores.
- Posee la capacidad de que los usuarios aprendan rápida y fácilmente a utilizarla y manipularla sin ningún tipo de impedimentos ni problemas.
- Mensajes claros y entendibles por los usuarios.

Debilidades del producto de software:

- Desde el punto de vista de la *personalización de los elementos de la interfaz de usuario*, la aplicación no permite modificarlos. Esto hace que al usuario no le agrade en cuanto a estética y no quede satisfecho con la CargoX.
- En cuanto a accesibilidad, más allá de su rango obtenido, es sólo un idioma que se encuentra definido en la descripción del producto (Portugués) y es un idioma también el realmente soportado (también Portugués). Con esto se deduce que sólo personas que hablan ese idioma pueden entender y utilizar fácilmente la aplicación móvil.
- Los usuarios no tienen la posibilidad de personalizar las funciones que posee la app.

##### 9.5.4.1.- Acciones de mejora propuestas

Teniendo en cuenta el resultado del proceso de evaluación del producto de software y la etapa de *Check (Verificar)* del proceso de mejora continua, detallado en la sección 2.4.- *Proceso de mejora continua* se proponen a continuación una serie de acciones con el fin de mejorar las debilidades mencionadas en la sección anterior. Además esta sección de propuesta de acciones de mejora se corresponde con la etapa *Act (Actuar)* del proceso de mejora continua.

A continuación se detallan estas acciones de mejora propuestas mencionadas con anterioridad:

- Agregar funcionalidades a la aplicación móvil que puedan personalizarse, o bien cambiar las ya existentes para que los mismos puedan manipularlas y modificarlas a su gusto.

- Agregar más cantidad de idiomas a ser soportados por CargoX para que usuarios de otros países puedan utilizar y entender la aplicación para transportar cargas inclusive hacia otros países que manejan otras lenguas.
- Permitir a los usuarios personalizar las funciones que posee la aplicación, o bien agregar nuevas funciones que puedan ser modificadas.
- De la totalidad de mensajes que posee la CargoX, lograr una mayor cantidad que sean entendibles y claros para los usuarios, ya que es un punto importante para lograr que la app pueda ser más operable y controlable.

Con estas acciones propuestas se pretende resolver las debilidades detectadas en el producto de software durante la evaluación, consolidar las fortalezas del mismo, y establecer un procedimiento formal de pruebas y evaluaciones, las cuales son realizadas por el equipo de testing de CargoX. Poseer un equipo de este tipo proporciona las siguientes ventajas:

- Al emplear el 100% del tiempo para testing, se puede realizar pruebas o evaluaciones más detalladas y de mayor cobertura, aislando el trabajo de este equipo con el resto (como desarrollo por ejemplo).
- Evaluaciones y conclusiones más objetivas ya que sólo se encargan de las pruebas o evaluaciones, y no del desarrollo, diseño, etc.

La formalización de las pruebas del producto, así como la realización de evaluaciones periódicas de calidad, y las revisiones técnicas formales, favorecerán el incremento del nivel de calidad no solo del producto en sí sino del proceso de desarrollo.

Finalmente las mejoras propuestas deberán ser planificadas y luego implementadas, estas dos acciones corresponden a las etapas de *Plan (Planificar)* y *Do (Hacer)* del proceso de mejora continua definido en la sección 2.4.- *Proceso de mejora continua* y serán evaluadas en otra iteración de este.

### 9.6.- Conclusiones

En esta sección se expusieron los resultados de la evaluación del producto de software, junto con los comentarios, observaciones y conclusiones. Junto a cada métrica, subcaracterística y característica de calidad fue necesario indicar el rango, el cual se obtiene de la manera que se indica en la sección 6.3.- *Niveles de rating*. También se comentaron los resultados, ya sea aspectos a mejorar o destacables, para poder llegar a conclusiones parciales y luego a una conclusión final de calidad, mencionando luego las fortalezas y debilidades que posee el producto de software y a partir de estos proponer acciones de mejora para implementar en siguientes iteraciones del proceso de mejora continua.

El propósito de la evaluación definido en la sección 5.1.- *Propósito de la evaluación* se cumplió: se obtuvo el nivel de calidad del producto de software, identificando sus debilidades y fortalezas de una manera formalmente medible, en el marco de un proceso de mejora continua. La etapa *Check (Verificar)* de este proceso es donde se enmarcó el proceso de evaluación, y la propuestas de acciones de mejora del presente reporte se enmarcan en la etapa *Act (Actuar)*. Para completar dicho proceso restan como próximos pasos que exceden al presente trabajo. Estos son *Plan (Planificar)* y *Do (Hacer)*, ambos mencionados en la sección 2.4.- *Proceso de mejora continua*.

---

## CAPITULO 10.- CONCLUSION

---

En el presente trabajo se realizó una investigación teórica sobre calidad del producto de software; luego se ahondó el concepto de calidad en aplicaciones móviles, profundizando más en el tema de “app’s” móviles, que ayudó a clarificar la tarea de realizar un proceso de evaluación de calidad sobre una de ellas.

A lo largo del *CAPITULO 2.- CONCEPTOS DE CALIDAD EN EL SOFTWARE* se presentó la definición de calidad, la importancia que tiene la calidad en el software, entrando más en detalle en la calidad del producto y no tanto en la calidad del proceso. Se estudió la familia de normas internacionales ISO/IEC 25000 (SQuaRE).

En el *CAPITULO 3.- CALIDAD EN APLICACIONES MOVILES* se investigó acerca de la calidad en las aplicaciones móviles, se realizó una definición de las mismas, su contexto, pruebas de software y el desarrollo de las “app’s”. El objetivo de este capítulo fue introducir el tema de aplicaciones móviles.

En el *CAPITULO 4.- DESCRIPCION DEL PRODUCTO A EVALUAR* se describió el producto de software a evaluar. Junto a esto se definió el alcance, objetivos, su arquitectura, su utilidad en el mercado y para su posterior evaluación de calidad, una descripción aún más detallada del producto.

A partir del *CAPITULO 5.- REQUISITOS DE LA EVALUACION* se inician las etapas que conforman el proceso de evaluación de calidad según la norma *ISO/IEC 25040 – Evaluation process*. Durante este capítulo se definieron cuestiones muy importantes, como el propósito de la evaluación, motivación, tipo de producto a evaluar, el modelo de calidad a utilizar, se seleccionaron las características que se evalúan posteriormente, se presentó la importancia que tienen estas características y sus subcaracterísticas elegidas, se definieron los requerimientos de calidad, un grado de rigurosidad alto.

En el *CAPITULO 6.- ESPECIFICACION DE LA EVALUACION* se definió el alcance de la evaluación. Se seleccionaron las métricas a aplicar para evaluar cada requerimiento de calidad. Aquellas métricas fueron escogidas teniendo en cuenta, por sobre todas las cosas, la opinión de los desarrolladores de la “app” y, además, los datos que la empresa Sontra podía proporcionar (por temas de confidencialidad de la empresa) para poder llegar a realizar los cálculos necesarios. Este punto acotó el trabajo ya que algunas de las métricas existentes en [1] fueron descartadas dejándolas para una posterior evaluación, ajena a la tesina, por no contar con la información necesaria. Se definieron, también, los niveles de rating. Finalmente se estableció el criterio de evaluación para determinar el nivel de rating alcanzado por cada subcaracterística, característica y por el producto en sí, en función del alcanzado por las métricas.

En el *CAPITULO 7.- PLAN DE EVALUACION* se mencionaron objetivos, alcances, descripción del producto a evaluar, requerimientos de la evaluación, características de calidad aplicables lista de prioridades y objetivos de calidad.

La ejecución de la evaluación se presentó en el *CAPITULO 8.- EJECUCION DE LA EVALUACION*.

Finalmente, y terminando el proceso de evaluación de calidad, se mostraron los resultados obtenidos durante todo el *CAPITULO 9.- REPORTE DE LA EVALUACION*, junto con comentarios.

En conclusión, el objetivo pautado en este trabajo se considera cumplido. Gracias a todo el proceso de evaluación de calidad realizado a lo largo de esta tesina

se pudieron detectar las ventajas o las fortalezas, y las desventajas o debilidades del producto de software elegido, el cual es una aplicación móvil real y en funcionamiento actualmente, para así proponer acciones de mejora y poder promover un proceso de mejora continua, para que, en los siguientes ciclos de este proceso vaya aumentando de forma gradual la calidad, en este caso, de CargoX.

Gracias a este trabajo se obtuvo el conocimiento necesario para realizar la evaluación de calidad sobre una aplicación móvil, estudiando la familia de las normas internacionales ISO/IEC 25000 que son las que hoy en día se utilizan para evaluar la calidad de un producto de software, y utilizando métricas externas que se adaptan a las “app’s”.

Más allá de que el mundo de las aplicaciones móviles esté creciendo de manera exponencial, y teniendo en cuenta que las métricas que se exponen en la norma ISO/IEC 25023 [1] son para productos de software genéricos, se decidió no definir nuevas métricas, siendo suficiente con las que se utilizaron en el presente trabajo.

Sí, se deja abierta la discusión de generar nuevas métricas tomando las particularidades de las app’s y/o haciendo uso de otras normas como ISO/IEC 27001 (Information technology - Security techniques - Information security management systems - Requirements) específica para el aspecto de seguridad en informática.

Concluyendo, es importante resaltar que un producto de software de alta calidad genera:

- Un mayor número de clientes satisfechos, los cuales continúan adquiriendo el producto y probablemente se sientan inclinados a adquirir otros productos o servicios de la organización.
- Clientes dispuestos a recomendar a la organización a potenciales nuevos clientes.
- Aumento de la participación de la organización en el mercado.
- Organizaciones más estables.
- Beneficios en los empleados: orgullo en los productos realizados, satisfacción en un trabajo bien hecho.
- Menor esfuerzo en el mantenimiento del producto y menor esfuerzo en la asistencia al usuario, lo que implica que el proveedor del software puede invertir más tiempo en trabajo productivo.

Con esto se concluye que siempre es recomendable realizar o generar productos de software de alta calidad, por eso es tan importante realizar evaluaciones de calidad para poder saber si, en este caso, un producto es de alta calidad o no, y si no lo es, mejorarlo para así poder competir en el mercado y mejorar como organización en todos los aspectos.

---

## CAPITULO 11.- REFERENCIAS

---

[A] Ian Sommerville. *Ingeniería de Software*. Novena Edición. Addison-Wesley. 2011.

[B] Paulk. *Capability Maturity Model for Software*. 1994.

[C] R. Pressman, *Ingeniería de Software: Un Enfoque Práctico*. Séptima Edición. McGrawHill. 2010.

[D] Pesado, Bertone, Pasini, Esponda, Martorelli. "Calidad en el desarrollo de Sistemas de Software". X Workshop de Investigadores en Ciencias de la Computación. Mayo, 2008.

[E] Portal de World Quality Report. Fuente con fecha de acceso 01/05/2018:  
<https://www.capgemini.com/thought-leadership/world-quality-report-2016-17>

[F] Portal de la Norma ISO/IEC 25000. Fuente con fecha de acceso 15/04/2018:  
<http://www.iso25000.com/>.

[G] ISO/IEC 25010. *Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - System and software quality models*. 2011.

[H] ISO/IEC CD 25022.3  
*Systems and software Engineering –  
Systems and software Quality Requirements and Evaluation (SQuaRE) –  
Measurement of quality in use*. 2014.

[I] ISO/IEC CD 25023.3  
*Systems and software engineering –  
Systems and software Quality Requirements and Evaluation (SQuaRE) –  
Measurement of system and software product quality*. 2014.

[J] ISO/IEC 25040:2011  
*Systems and software Engineering –  
Systems and software Quality Requirements and Evaluation (SQuaRE) –  
Evaluation Process*.

[K] César Camisón. *Conceptos de calidad y enfoques de gestión*. 2009. Fuente con fecha de acceso 15/04/2018: <http://www.mailxmail.com/curso-conceptos-calidad-enfoques-gestion/dimensiones-calidad-producto>.

[L] Portal de la empresa Sontra. Fuente con fecha de acceso 15/04/2018:  
<https://cargox.com.br/>

[M] "Los retos de la calidad en aplicaciones móviles". Fuente con fecha de acceso 01/05/2018: <http://www.computing.es/movilidad/opinion/1101163046501/retos-calidad-de-aplicaciones-moviles.1.html>

[N] *"The importance of usability in mobile apps". Fuente con fecha de acceso 01/05/2018:* <https://knowtechie.com/the-importance-of-usability-in-mobile-apps/>

[O] *"The Importance of Application Security: A Few of the Benefits and Risks": Fuente con fecha de acceso 01/05/2018:* <https://www.veracode.com/blog/intro-appsec/importance-application-security-few-benefits-and-risks>

[P] Dave Zubrow. *Software Quality Requirements and Evaluation, the ISO 25000 Series. PSM Technical Working Group. February 2004. Fuente con fecha de acceso 15/04/2018:* <http://www.psmc.com/downloads/twgfeb04/04zubrowiso25000swqualitymeasurement.pdf>.

[Q] *La importancia de la calidad del software. Fuente con fecha de acceso 15/04/2018:* <https://www.baquia.com/emprendedores/la-importancia-de-la-calidad-del-software>.